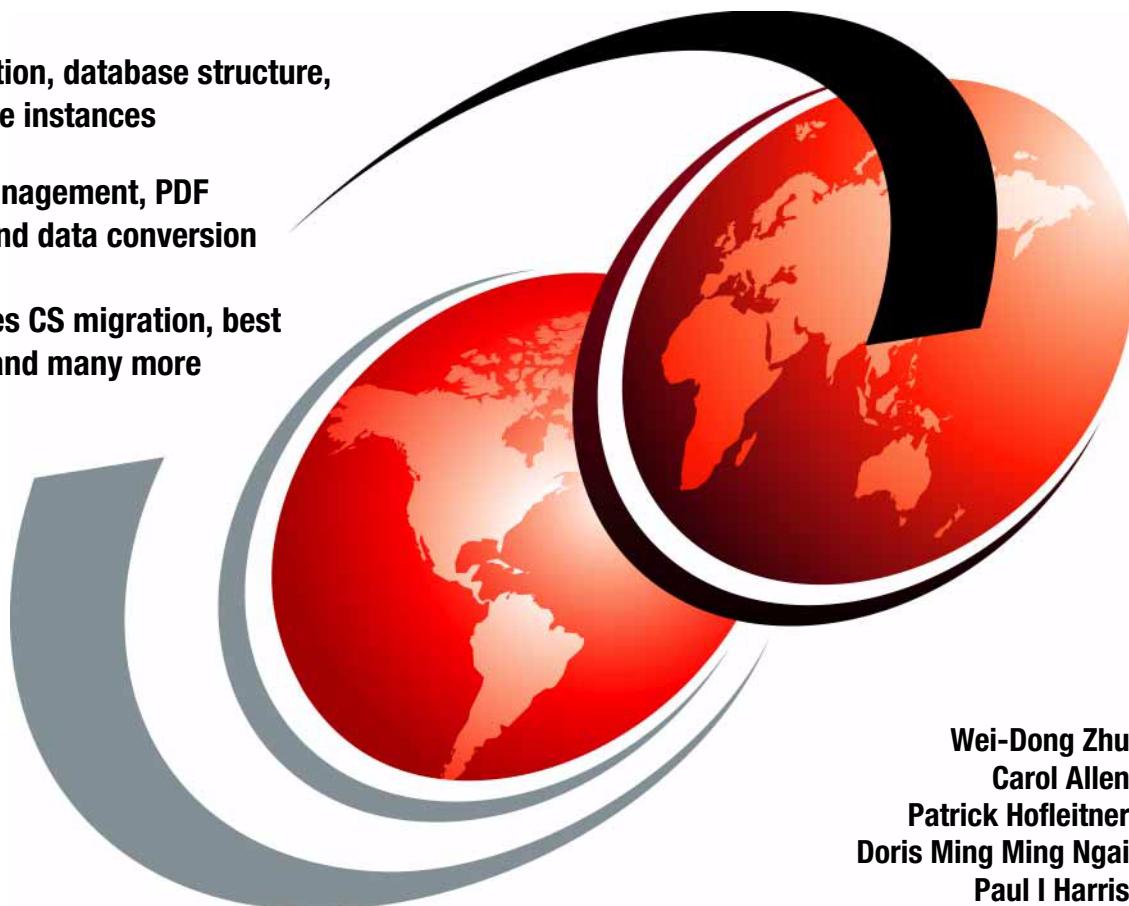


Content Manager OnDemand Guide

Administration, database structure,
and multiple instances

Storage management, PDF
indexing, and data conversion

Exits, iSeries CS migration, best
practices, and many more



Wei-Dong Zhu
Carol Allen
Patrick Hofleitner
Doris Ming Ming Ngai
Paul I Harris

Redbooks



International Technical Support Organization

Content Manager OnDemand Guide

December 2007

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

Third Edition (December 2007)

This edition applies to Version 7, Release 1, IBM DB2 Content Manager OnDemand for Multiplatforms (product number 5697-G34), Version 7, Release 1, IBM DB2 Content Manager OnDemand for z/OS and OS/390 (product number 5655-H39), and Version 5, Release 3, IBM DB2 Content Manager OnDemand for iSeries™ (product number 5722-RD1).

© Copyright International Business Machines Corporation 2003, 2006, 2007. 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

Figuresxi
Tablesxix
Noticesxxi
Trademarksxxii
Prefacexxiii
The team that wrote this IBM Redbooks publicationxxiv
Become a published authorxxvi
Comments welcome.....	.xxvi
Summary of changesxxix
December 2007, Third Editionxxix
May 2006, Second Editionxxx
Chapter 1. Overview and concepts	1
1.1 Overview of OnDemand	2
1.2 Concepts	4
1.2.1 Report and document	4
1.2.2 Application, application group, and folder.....	5
1.2.3 Indexing methods	8
1.2.4 OnDemand server and its components.....	9
1.3 Supported environments	15
1.4 Document access and distribution possibilities.....	17
1.4.1 Document access	17
1.4.2 Document distribution possibility.....	18
1.4.3 Federated search using WebSphere Information Integrator Content Edition.....	19
Chapter 2. Administration	21
2.1 Report administration	22
2.1.1 Storage sets	23
2.1.2 Application groups.....	23
2.1.3 Applications.....	30
2.1.4 Folders	35
2.1.5 The Report Wizard	43
2.2 User and group administration	47
2.2.1 User types, authorities, and functions	48

2.2.2 Decentralized system administration	49
2.2.3 OnDemand XML Batch Administration	53
Chapter 3. Database structure	63
3.1 System control tables	64
3.2 Main data table structures	66
3.3 Relationship between databases when loading data	69
3.4 Search sequence from folder	71
3.5 System log.	73
3.6 Database creation and relationship on z/OS.	73
3.6.1 System tables	73
3.6.2 Data tables	75
Chapter 4. Multiple instances	77
4.1 OnDemand instance	78
4.2 Multiple instances on UNIX	78
4.2.1 Defining a second instance	78
4.2.2 Working with the second instance.	84
4.3 Multiple instances on Windows NT	86
4.3.1 Defining a second instance	86
4.4 Multiple instances on iSeries.	86
4.5 Multiple instances on z/OS	91
4.5.1 Understanding file systems in UNIX System Services	93
4.5.2 Creating an instance on z/OS	97
Chapter 5. Storage management	105
5.1 Tivoli Storage Manager for Multiplatforms.	106
5.1.1 Tivoli Storage Manager overview	106
5.1.2 Configuring OnDemand for Tivoli Storage Manager archive management.	115
5.1.3 OnDemand storage management.	118
5.1.4 Storage set definition	119
5.1.5 Application group storage management	123
5.1.6 Advanced application group storage management.	125
5.1.7 OnDemand with IBM System Storage Archive Manager	126
5.1.8 The arsmaint command	131
5.2 Object access method for z/OS.	132
5.2.1 OAM components and SMS terminology	133
5.2.2 Defining a storage set	136
5.2.3 Storing data to Virtual Storage Access Method data sets.	140
5.3 Archive Storage Manager for iSeries	141
5.3.1 Migration policy	142
5.3.2 Application group storage management	152
5.3.3 Advanced application group storage management.	155

5.3.4 Advanced application group database information	156
Chapter 6. Performance	159
6.1 When performance tuning is necessary	160
6.2 Tuning OnDemand to enhance performance	162
6.2.1 OnDemand architecture	162
6.2.2 OnDemand configuration	163
6.2.3 Database	164
6.2.4 System management	166
6.2.5 Storage management	169
6.2.6 ODWEK configuration	170
6.3 Performance issues based on data type	177
6.3.1 PDF data	177
6.3.2 Line data	179
6.3.3 AFP data	180
Chapter 7. PDF indexing	185
7.1 Getting started	186
7.1.1 What is the Portable Document Format?	186
7.1.2 PDF and the OnDemand client	187
7.2 Indexing issues with PDF	188
7.2.1 Listing fonts in a PDF file	189
7.3 PDF indexer	190
7.3.1 PDF indexing on z/OS	194
Chapter 8. OnDemand Web Enablement Kit	199
8.1 ODWEK architecture	200
8.1.1 ODWEK access to OnDemand	200
8.2 Integrating with APIs	202
8.2.1 HTML samples (URL API)	203
8.2.2 Java API samples	206
8.3 OnDemand Portlets	216
8.3.1 Features and functions included in the OnDemand Portlets	216
8.3.2 Hardware and software requirements	217
8.3.3 Configuring and deploying the IBM OnDemand Portlets 3.2	218
8.3.4 Using OnDemand Portlets	220
8.4 Deploying the ODWEK servlet	225
8.4.1 Before you begin	226
8.4.2 Copying the files	226
8.4.3 Deploying the servlet using WebSphere tools	227
Chapter 9. Data conversion	249
9.1 Overview of data conversion	250
9.1.1 Why convert data streams	250

9.1.2 When to convert data streams	250
9.1.3 How to convert the data: integrated solutions with OnDemand	251
9.2 IBM AFP2WEB Services Offerings and OnDemand	252
9.2.1 AFP2HTML: converting AFP to HTML	253
9.2.2 AFP2PDF: converting AFP to PDF	261
9.2.3 AFP2XML: converting AFP to XML	272
9.2.4 AFPIindexer: indexing composed AFP files	275
9.3 Xenos and OnDemand	278
9.3.1 Converting AFP to XML through ODWEK	281
9.3.2 Using the AFP2PDF transform with ARSLOAD	297
9.3.3 Job Supervisor program	304
Chapter 10. Report Distribution	307
10.1 Introduction to Report Distribution	308
10.1.1 What documents are needed	310
10.1.2 Who will receive the documents	310
10.1.3 When will the documents be retrieved and delivered	310
10.1.4 Where will they be delivered	311
10.2 Defining the distribution	311
10.2.1 Adding a report	316
10.2.2 Adding a banner	319
10.2.3 Adding a schedule	320
10.2.4 Adding a bundle	321
10.2.5 Adding a distribution	324
10.2.6 Report Distribution parameters	328
10.3 Hints and tips	330
Chapter 11. Exits	335
11.1 Introduction to user exits	336
11.2 ACIF exits	337
11.2.1 Input record exit	337
11.2.2 Index record exit	339
11.2.3 Output record exit	341
11.2.4 Resource exit	342
11.2.5 Debugging user exit programs	342
11.3 System administration	344
11.3.1 System log exit for Multiplatforms	344
11.3.2 System log exit for z/OS	349
11.3.3 Print exit for Multiplatforms	354
11.4 Customized functions	360
11.4.1 The user exit header file arscsxit.h	361
11.4.2 Fax options exit	364
11.4.3 Load exit	366

11.4.4 Permission exit	367
11.4.5 Client retrieval preview exit	371
11.4.6 Security exit	373
11.4.7 Security exit on z/OS.	376
11.4.8 ARSYSPIN and sample APKACIF exit on z/OS	386
11.4.9 Storage management external cache exit.	393
11.4.10 Tablespace creation exit	394
Chapter 12. iSeries Common Server migration	399
12.1 Introduction to the migration tool	400
12.2 Preparation	400
12.2.1 Setting up the Common Server environment	400
12.2.2 Making changes to the Spool File Archive environment	402
12.3 Analysis and planning	404
12.4 Migration	410
12.4.1 Migration without the tool	410
12.4.2 Migration with the tool	420
12.5 Modifications to folders after migration	423
12.6 Ongoing use of the Common Server.	427
12.7 Summary	427
Chapter 13. Solution design and best practices	429
13.1 Designing a winning solution.	430
13.1.1 What is OnDemand?	430
13.1.2 Three-step approach in solution design	432
13.1.3 Solution design case study	436
13.2 Best practices	439
13.2.1 Including a Date field in an application group	440
13.2.2 Including a Load Date field in an application group	441
13.2.3 Including an Application ID Field in an application group	443
13.2.4 Application group name handling	446
13.2.5 PDF document access	449
13.2.6 PDF document indexing	451
Chapter 14. Troubleshooting	453
14.1 Troubleshooting FAQ	454
14.1.1 Determining the nature of the problem	454
14.1.2 Indexing or loading issue	454
14.1.3 OnDemand maintenance	458
14.1.4 OnDemand startup problem	459
14.1.5 OnDemand client issue	460
14.1.6 ODWEK matter	461
14.2 Information collection	461
14.2.1 Indexing or loading	463

14.2.2 Database	465
14.2.3 Tivoli Storage Manager	467
14.2.4 OnDemand client logon	467
14.2.5 Performance	468
14.2.6 ODWEK	468
14.2.7 OnDemand server hang or crash	469
14.2.8 Exporting information to a local server	469
14.3 OnDemand trace facility	474
14.3.1 Enabling the trace facility	474
14.3.2 Setting trace parameters in the OnDemand administrative client .	477
Chapter 15. Did you know	479
15.1 Using the Document Audit Facility	480
15.1.1 Creating a status field in the application group	481
15.1.2 Setting a default in the application	482
15.1.3 Updating permissions for various users	483
15.1.4 Creating folders for each user type	484
15.1.5 Creating the Document Audit Facility control file	485
15.1.6 Viewing the folders	486
15.1.7 Tracking changes to invoice status	487
15.2 Related Documents feature	489
15.2.1 How it works	490
15.2.2 Configuring Related Documents	491
15.3 Store OnDemand	493
15.3.1 Why it is needed	494
15.3.2 What it does	495
15.4 OnDemandToolbox	496
15.5 Batch OnDemand commands in z/OS	499
15.6 Testing PDF indexer in z/OS	500
15.7 Date range search tip for users	501
15.8 Ad-hoc CD-ROM mastering	502
15.8.1 Transferring documents from the OnDemand server to the staging drive	503
15.8.2 Burning the disk image to the optical media	508
15.8.3 Accessing the CD image from disk	508
15.8.4 Accessing the CD image from the CD-ROM	509
15.9 OnDemand Production Data Distribution	511
15.9.1 Why it is needed	511
15.9.2 How it works	511
15.10 Customizing the About window	511
15.11 Modifying client behavior through the registry	513
15.11.1 Single-selection hit list	514
15.11.2 Folder List Filter enhanced	515

15.11.3 Customizing your line data background	517
15.11.4 Displaying the OnDemand splash screen or About window	519
15.12 Negative numbers in decimal fields handling	520
15.13 Message of the day	522
15.14 OnDemand bulletins	524
Chapter 16. Optional features	525
16.1 OnDemand Distribution Facility (ODF) on z/OS	526
16.1.1 Additional features	527
16.1.2 ODF administration process	528
16.1.3 Updated documentation and APAR lists	544
16.2 Report Distribution	554
16.2.1 Report Distribution components	555
16.2.2 Setting up Report Distribution	557
16.3 Content Manager OnDemand Toolbox	568
16.3.1 OnDemand Toolbox installation	569
16.3.2 OnDemand Toolbox components	569
16.4 E-mail Notification and Delivery for Multiplatforms	570
16.4.1 E-mail Notification and Delivery sample output	571
Chapter 17. Enhancements	573
17.1 Web Administration Client	574
17.1.1 Requirements and limitations	577
17.2 Composite indexes	578
17.2.1 Add a composite index	578
17.3 Cluster indexes	583
17.3.1 Define a cluster index	584
17.4 Cabinets	590
17.4.1 Cabinet authorities	591
17.5 File name indexing	592
17.5.1 Set up file name indexing	593
17.6 LDAP security	599
17.6.1 Enable and disable LDAP authentication	600
17.7 64-bit support	602
17.8 Tracing	602
17.8.1 Configure tracing	602
Appendix A. Additional material	605
Locating the Web material	605
Using the Web material	606
System requirements for downloading the Web material	606
How to use the Web material	606
Glossary	607

Related publications	633
IBM Redbooks	633
Other resources	633
Referenced Web sites	635
How to get IBM Redbooks	636
IBM Redbooks collections	636
Index	637

Figures

1-1	OnDemand system overview	3
1-2	Applications and documents	5
1-3	The concepts of folders, application groups, and applications	7
1-4	Examples of folders, application groups, and applications	8
1-5	Data indexing and loading process	14
2-1	OnDemand system components relationship	22
2-2	Database information	24
2-3	Application group storage management	25
2-4	Application group Field Information tab	27
2-5	Application load information	31
2-6	Indexer Properties for PDF graphical Indexer	34
2-7	Folder general information	36
2-8	Folder permissions	38
2-9	Folder field definition: Text Search field	39
2-10	Folder Permissions with Full Report Browse for Administrative Client .	41
2-11	Folder view with Full Report Browse authority on a Windows client .	42
2-12	Report Wizard button on the OnDemand Administrator Client	43
2-13	An Application identifier Windows with Report Wizard	44
2-14	Names page of the Report Wizard window	45
2-15	Names page of Report Wizard window when adding an application .	46
2-16	Decentralized system administration: object type model	49
2-17	Decentralized system administration: object owner model	51
3-1	Application group identifier example	67
3-2	Relationship between system tables and data tables	70
3-3	OnDemand client search criteria panel	71
3-4	Search sequence from a folder	72
3-5	Relation between configuration files, tablespace, and system tables .	74
4-1	ARS.INI file sample	80
4-2	ARS_ONTEST.CFG file sample	82
4-3	ARS.INI file sample for the iSeries server	89
4-4	ARS.CFG file sample for the iSeries server	90
4-5	Client server setup for the iSeries server	91
4-6	Multiple instances overview on z/OS	92
4-7	JCL used to allocate an HFS file in z/OS	93
4-8	OnDemand file structure in UNIX System Services	96
4-9	Mount of a file system	97
4-10	The arsins1.cfg file for the new instance on z/OS	99
4-11	The ARS.INI file for two instances	100

4-12	Starting a second instance	102
4-13	Relationship between the steps for creating an instance on z/OS	103
4-14	ARSLOAD for new instance	104
5-1	Tivoli Storage Manager storage objects	107
5-2	Storage pool hierarchy	111
5-3	Client node configuration	112
5-4	Backup archive client	114
5-5	Restore client	115
5-6	Windows configurator	116
5-7	ARS.CFG Tivoli Storage Manager configuration	117
5-8	OnDemand storage objects	118
5-9	Storage set definition	120
5-10	Primary node definition	121
5-11	Application group storage management	123
5-12	Advanced application group storage management	125
5-13	Adding a storage set	136
5-14	Adding a primary storage node to the storage set	137
5-15	Relationship between OAM and OnDemand	138
5-16	Defining index expiration in OnDemand	139
5-17	Defining a storage set for VSAM	140
5-18	iSeries Access Navigator	145
5-19	iSeries Disk Pool definition	145
5-20	iSeries optical storage group	146
5-21	iSeries optical volume	147
5-22	iSeries migration policy	148
5-23	iSeries new policy level	150
5-24	iSeries migration policy hierarchy	152
5-25	iSeries application group storage management	153
5-26	iSeries application group advanced storage management	155
5-27	Single table for all loads	157
6-1	The arsload process	160
6-2	The OnDemand architecture	162
6-3	The ODWEK cache	172
6-4	ODWEK with multiple user ID access	175
6-5	ODWEK with single user ID access	176
6-6	PDF architecture	178
6-7	Transaction data in the graphical indexer	180
6-8	Collecting AFP fonts	181
6-9	Disabling compression	182
7-1	Listing fonts in a PDF document	189
7-2	Error message if PDF does not view	191
7-3	Capturing text with the PDF graphical indexer	192
7-4	Sample PDF font mapping table as PDS	195

8-1	ODWEK with CGI, ArsWWWServlet, and a custom servlet	202
8-2	Web page of a sample integrated ODWEK installation	203
8-3	Logon panel in the OnDemand Portlet	222
8-4	Folder list in the OnDemand Portlet	222
8-5	Search panel in the OnDemand Portlet	223
8-6	Hit list in the OnDemand Portlet	223
8-7	Image document displayed by the OnDemand Viewer portlet	224
8-8	AFP document displayed by the OnDemand Viewer portlet	224
8-9	Line data document displayed in OnDemand Portlet	225
8-10	WebSphere Application Server Toolkit workbench	229
8-11	Creating a new project	230
8-12	J2EE Specification version panel	231
8-13	Enterprise Application Project panel	232
8-14	EAR Module Projects panel	233
8-15	New Module Project panel	234
8-16	EAR Module Projects panel	235
8-17	Confirm Perspective Switch window	235
8-18	WebSphere Application Server Toolkit J2EE perspective	236
8-19	Import Class Files panel	237
8-20	Importing class files from the local file system	238
8-21	Web deployment Descriptor editor	239
8-22	Servlets tab	240
8-23	Adding a servlet	241
8-24	Adding a servlet	242
8-25	URL mapping	243
8-26	Initialization section	244
8-27	ConfigDir parameter value	245
8-28	Web Deployment Descriptor save changes confirmation message	246
8-29	EAR Export display	247
9-1	AFP2PDF and AFP2HTML	252
9-2	AFP invoice	273
9-3	Defining an XML ICT file for AFP invoices	274
9-4	Invoice XML file	274
9-5	AFP invoice	276
9-6	Defining an XML ICT file for AFP invoices	277
9-7	AFP document tagged with TLEs for OnDemand indexation	278
9-8	How the Xenos transforms fit into the OnDemand environment	280
9-9	AFP billing statement stored in OnDemand	282
9-10	AFP document presented in the XML format sample	292
9-11	AFP document displayed in its native format with AFP Web Viewer	293
9-12	Sample document output using XML and CCS	294
9-13	Application indexing parameters for Xenos	298
9-14	Job Supervisor program syntax	305

10-1	List of monthly sales reports for Comp A	309
10-2	OnDemand Administrator Client with Report Distribution	311
10-3	Bundle contents	313
10-4	User with an e-mail address and server printer specified	315
10-5	Report window for the Northwest regional sales report	317
10-6	SQL Query window for the Northwest regional sales report	318
10-7	Add a Banner window	319
10-8	Example of a separator banner	320
10-9	Add a Schedule window	321
10-10	General tab of the Add a Bundle window	322
10-11	Bundle Contents tab of the Add a Bundle window	323
10-12	General tab of the Add a Distribution window	324
10-13	Bundle tab of the Add a Distribution window	325
10-14	Schedule tab of the Add a Distribution window	326
10-15	Recipients tab of the Add a Distribution window	327
10-16	Report Distribution Parameters window	329
10-17	Search for Distributions window	332
11-1	Select OnDemand system parameters	345
11-2	System Parameters window for user exit select	349
11-3	Setting the query restriction	386
11-4	Indexer Properties window	391
11-5	Specify the exit load module name	392
12-1	Defining the RTPS01 query	411
12-2	Specify File Selections display	411
12-3	Define Result Fields display	412
12-4	Select and Sequence Fields display	412
12-5	Select Records display	412
12-6	Output type	412
12-7	Database output	413
12-8	Running query RPTS01	413
12-9	PRTRPTS01 CL program	414
12-10	Application version	415
12-11	Update an Application display	415
12-12	Update an Application Group	416
12-13	Adding an application ID value	417
12-14	DLTRPTS01 CL program	418
12-15	Folder settings before making changes	423
12-16	Folder settings after making changes	423
12-17	Default settings for updating the folder date	424
12-18	Adding a folder	425
12-19	Updating the FINRPTS folder	425
12-20	Update application group	426
12-21	Folder search	426

13-1	InvoiceDate as segment date	440
13-2	Message indicating no date field chosen as a segment	441
13-3	Specifying a load date	442
13-4	Application ID Field	444
13-5	Adding an Application with a version ID.	445
13-6	Defining an Application Group field	447
13-7	The Application Group name displayed in OD client	448
13-8	Error message: Application ID value used as identifier in application .	449
13-9	Indexer Information tab	449
13-10	View Information tab	450
14-1	Document Properties window for the PDF document	457
14-2	Error message from the console	459
14-3	Setting up the local server	470
14-4	Add a Server window	471
14-5	Setting up the local server	471
14-6	Exporting an application group	472
14-7	Export Application Group window	473
14-8	Enabling Trace Parameters from the administrative client	477
14-9	System Trace Settings window	478
15-1	Adding status field to the application group	481
15-2	Setting default in application	482
15-3	Query restriction	483
15-4	Application group permission: no query restriction	484
15-5	ARSGUI.CFG.	485
15-6	Auditors view of the folder	486
15-7	Customer view of the folder	487
15-8	Message logging setup	488
15-9	Field update logging setup.	488
15-10	Related Documents search	491
15-11	Registry structure for Related Documents feature.	493
15-12	Store OnDemand screen display.	495
15-13	BPXBATCH sample program.	499
15-14	ARSPDOCI sample JCL	500
15-15	Client installation: selecting the Ad-Hoc CDROM Mastering option. . .	502
15-16	Set CD-ROM Mastering Options	503
15-17	Set CD-ROM Mastering Options: staging path	503
15-18	Starting the CD-ROM Mastering feature	504
15-19	CD-ROM Mastering: adding folders to CD-ROM.	505
15-20	CD-ROM Mastering process	506
15-21	CD-ROM Mastering: process complete message	506
15-22	CD-ROM Mastering: adding another folder	507
15-23	Accessing a CD image from disk: updating servers	508
15-24	Accessing CD image from disk: log in	509

15-25 Accessing CD image from disk: opening a folder	509
15-26 Accessing CD image from CD-ROM: logging in	510
15-27 Accessing CD image from CD-ROM: opening a folder	510
15-28 Customized About menu	512
15-29 Customized About window.	513
15-30 Registry setting modification: single-selection hit list	514
15-31 Registry setting modification: enhanced Folder List Filter option.	515
15-32 Registry setting modification: FILTER_AUTO_APPEND_WILDCARD	515
15-33 Registry setting modification: FILTER_AUTO_UPPERCASE	516
15-34 Registry setting modification: Folder List Filter changes	516
15-35 Line data background customization: Green Bar Alternating Stripes .	517
15-36 Editing colors	518
15-37 Registry setting: GREEN_BAR_COLOR	518
15-38 Output of registry setting for GREEN_BAR_COLOR.	519
15-39 Registry setting: SHOW LOGO	520
15-40 Negative number capture in graphical indexer	521
15-41 Negative number displayed in OnDemand client.	521
15-42 Message of the day	522
15-43 Windows registry for Message of the day	523
16-1 ODF Monitor	529
16-2 ODF Administration screen	529
16-3 Maintain Recipient screen	530
16-4 Maintain recipient list screen - Display recipient information	531
16-5 Display Recipient: Display recipients defined in ODF database	531
16-6 Cross Reference Maintenance screen.	532
16-7 Maintain Distribution screen - With option 5	533
16-8 Maintain Distribution screen - With option 6	534
16-9 Display Distributions screen.	535
16-10 Display Bundle Definition screen	536
16-11 Display Bundle Definition screen - with predefined fields	537
16-12 Maintain Bundle Definitions screen sample.	538
16-13 Bundle Definition Report Date Ranges screen	539
16-14 Maintain Bundle Definition: e-mail notification and delivery options .	543
16-15 Defining a report	558
16-16 Adding a report	559
16-17 Adding a banner.	560
16-18 Creating a bundle.	561
16-19 Defining a bundle	562
16-20 Defining a schedule	563
16-21 Adding a schedule	564
16-22 Defining a distribution	565
16-23 Defining a distribution name	566
16-24 Defining a distribution bundle.	567

16-25 Defining a distribution schedule	567
16-26 Adding a recipient	568
17-1 Web admin server login window	574
17-2 Web admin login window	575
17-3 Web administration window	575
17-4 Exploring existing users	576
17-5 User list	576
17-6 Copying user	577
17-7 Viewing an existing composite index	579
17-8 Updating an existing index	579
17-9 Composite index - Update Application Group, Advanced Index tab . .	580
17-10 Adding multiple indexes	581
17-11 Showing added composite index	582
17-12 Shows indexes that have been added	583
17-13 creating a new application group	584
17-14 Adding a new application group	585
17-15 Adding a storage name	585
17-16 Adding a database field name	586
17-17 Selecting a string value	586
17-18 Defining a database value	587
17-19 Adding a cluster index	588
17-20 Multiple field indexes panel	589
17-21 Cabinet to folder relationship	590
17-22 Viewing contents of cabinet	591
17-23 Cabinet Access and Administrator authority	592
17-24 Specify format of the date	593
17-25 Specify the format of the date	594
17-26 Specify the field information for the date field	595
17-27 Map the folder fields to the application group fields	595
17-28 Document search hit list	598
17-29 Accessing system parameters	600
17-30 Enabling LDAP authentication	601
17-31 Sample ARS.CFG to configure LDAP authentication	601
17-32 Selecting trace parameters	602
17-33 Selecting system trace settings	603
17-34 Trace settings file	604
17-35 Sample log file	604

Tables

2-1	Administrator roles in object type model	50
2-2	Administrator roles in object owner model	52
2-3	Administration guidelines	53
3-1	OnDemand system control tables	64
3-2	ARSSEG table structure	66
3-3	AG_internal_id table structure	68
3-4	ARSLOAD table structure	69
4-1	Permission byte information.	95
4-2	Subdirectories of /usr/lpp/ars	95
5-1	Comparison of Tivoli Storage Manager expiration methods with data protection OFF or ON	128
6-1	Performance troubleshooting reference.	161
6-2	Cached data description	171
7-1	Space allocation for font mapping table.	195
7-2	Data control block used for ADOBERES data set	197
7-3	DCB used for the ADOBEFNT data set.	197
7-4	DCB used for the INPUT data set	198
8-1	Copy of the shared library according to the operating system.	227
8-2	Copy of the ODWEK INI files according to the operating system	227
8-3	WebSphere Application Server Toolkit start command by operating system 228	
8-4	ConfigDir parameter value according to the operating system	245
9-1	Available Xenos transforms: transforming data at load time	279
9-2	Available Xenos transforms: transforming data dynamically through ODWEK	279
9-3	Parameter file summary.	284
11-1	User exits module	360
11-2	Security exit modules.	377
13-1	Local library compared with OnDemand	431
14-1	Determine the version of OnDemand in Multiplatforms	462
14-2	Information to collect for loading	463
14-3	Information to collect for common AFP problems	464
14-4	Information to collect for DB2.	465
14-5	Information to collect for Tivoli Storage Manager	467
14-6	Information to collect for client logon problems	467
14-7	Information to collect for performance issues	468
14-8	Information to collect for ODWEK	468
15-1	T format string	501

15-2	T format string examples with the Equal To search operator	502
15-3	T format string examples with the Between search operator.	502
16-1	ODF distribution tables	543

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

AIX®	Lotus®	S/390®
AS/400®	MVS™	System i™
CICS®	Notes®	System Storage™
DB2 Universal Database™	OS/2®	System/370™
DB2®	OS/390®	Tivoli®
DFSMSdfp™	OS/400®	TotalStorage®
Enterprise Storage Server®	Print Services Facility™	WebSphere®
eServer™	pSeries®	Workplace™
IBM®	Redbooks®	Workplace Forms™
IMS™	Redbooks (logo)  ®	z/OS®
iSeries®	RACF®	zSeries®
Lotus Notes®	REXX™	

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

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

PostScript, Distiller, Adobe, Acrobat, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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

Excel, Expression, Internet Explorer, Microsoft, Visual Basic, Windows NT, Windows Server, Windows Vista, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Pentium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

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

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

Preface

This IBM® Redbooks® publication covers a variety of topics relating to the practical application of IBM DB2® Content Manager OnDemand (simply referred to as “OnDemand”) for Multiplatforms Version 8.3 (also known as Version 7.1.2.5), z/OS® Version 7.1, and IBM eServer™ iSeries® Common Server Version 5.3 of the OnDemand product. Where necessary, separate sections are included to cover variations between the different platforms.

This IBM Redbooks publication provides helpful, practical advice, hints, and tips for those involved in the design, installation, configuration, system administration, and tuning of an OnDemand system. It covers key areas that are either not well known to the OnDemand community or are misunderstood. We reviewed all aspects of the OnDemand topics and decided to provide information about the following areas:

- ▶ Administration
- ▶ Database structure
- ▶ Multiple instances
- ▶ Storage management
- ▶ Performance
- ▶ PDF indexing
- ▶ OnDemand Web Enablement Kit
- ▶ Data conversion
- ▶ Report distribution
- ▶ Exits
- ▶ iSeries Common Server migration
- ▶ Solution design and best practices
- ▶ Troubleshooting
- ▶ Did you know?
- ▶ Option features
- ▶ Enhancements

Because a number of other sources are available that address various subjects on different platforms, this IBM Redbooks publication is not intended as a comprehensive guide for OnDemand. We step beyond the existing OnDemand documentation to provide insight into the issues that might be encountered in the setup and use of OnDemand.

Note: IBM DB2 OnDemand for Multiplatforms Version 8.3 is also known as Version 7.1.2.5 or later. This IBM Redbooks publication covers features and functions up to Version 7.1.2.5 of OnDemand for Multiplatforms.

The team that wrote this IBM Redbooks publication

This IBM Redbooks publication was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.

Wei-Dong Zhu is a Content Management Project Leader with the ITSO in San Jose, California. She is a Certified Solution Designer for IBM DB2 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.

Carol Allen is the President of Carol Allen Consulting, LLC, specializing in implementations, training, and migration services for customers using OnDemand for iSeries. She is a Certified Solution Expert for OnDemand for iSeries. Before starting her own business, she worked for IBM for 24 years on iSeries and IBM AS/400® document archive solutions since 1992. She has extensive experience in OnDemand customer implementations and consulting, along with writing and teaching technical classes around the world. Carol holds a Bachelor of Arts degree in psychology from Emory University and a Master in Counseling degree from Georgia State University.

Patrick Hofleitner is an IT Specialist at IBM France with more than 10 years of experience in the Content Management field. He is presently part of the Channels Technical Sales team, enabling, educating and supporting Business partners in Content Management field. He is a Certified Solution Expert for OnDemand for Multiplatforms and Ondemand for iSeries, and is a Certified Solution Designer for Content Manager V8. He has worked as a presales technical support and services specialist, selling, designing and implementing IBM Content Management solutions on diverse platforms. Patrick holds an Industrial Chemistry Engineering degree from INSA in Rouen, France.

Doris Ming Ming Ngai is an IT Availability Specialist for IBM Singapore. She has been working in Integrated Technology Services for eight years, in the IBM eServer pSeries® services and support team, spending most time in implementing and supporting OnDemand on AIX® and Microsoft® Windows®. Her areas of expertise include AIX, OnDemand, and Tivoli® Storage Manager. Doris is a Certified Solution Expert for OnDemand Multiplatforms. She holds a degree in electrical engineering from National University of Singapore.

Paul I Harris is a Software Specialist for IBM Content Manager and IBM Content Manager OnDemand on Windows platform within Global Technology Services. Paul supports Content Manager and Content Manager OnDemand across the EMEA Software Support Domain. He has supported the Content Management software family for about three years. Paul has a BTEC National Diploma in Computer Studies. He joined IBM in 2001.

Special thanks to the following people who co-authored the first version of this IBM Redbooks publication:

Mike Adair
Stephanie Kiefer Jefferson
Henry Martens
Martin Pepper

Special thanks to the following people for their contributions to this IBM Redbooks publication:

- ▶ *Debbie Wagner* for her Report Distribution, and XML Batch Administration materials
- ▶ *Benjamin Boltz* and *Eric Hann* for their solution design (for performance and user satisfaction) materials
- ▶ *Sebastian Welter* for his development of the free, open-source software, OnDemand Toolbox, and associated contributors *Peter Seckinger* and *Hans-Joachim Hein*.
- ▶ *Andy J. Smith* and *Simon E. Moore* for their development of the Store OnDemand application and the architects of the application, *Martin Pepper* and *Chris Corfield*
- ▶ *Darrell Bryant* for reviewing and correcting the IBM Redbooks content
- ▶ *Steve Henrickson* for his continuous support of OnDemand IBM Redbooks projects

Thanks to the following people who made this project possible:

Nelson Chen
Trang K Duong
Bill Fusaro
Christopher Lewis
Bob Lichens
John Link
Rosalba Lucero
Robert Marshall
Paula Muir
Kenneth Nolan
Sandi Pond

Kevin Van Winkle
Timothy Yeh
IBM Software Group, Boulder, U.S.

Terry Brown
Manimala Kumaravel
Nancy O'Brien

Mike Reece
Yewen Tang
IBM Software Group and IBM Sales and Distribution, U.S.

Craig Brossman
Ronald Smith
Dave Tanigawa
IBM Systems and Technology Group, Boulder and Poughkeepsie, U.S.

Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review IBM Redbooks publication form found at:

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:
redbook@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

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 SG24-6915-02
for Content Manager OnDemand Guide
as created or updated on May 25, 2012.

December 2007, Third Edition

While producing another IBM Redbooks publication, *Implementing Content Manager OnDemand Solutions with Case Studies*, SG24-7511, in 2007, we produced IBM Redbooks materials related to the optional features of Content Manager OnDemand and the enhancements of the product up to that time. Because these materials are generic to the entire Content Manager OnDemand product and are not solution implementation specific, we decided to include the materials produced from that book to this book instead.

New chapters

- ▶ Chapter 16, “Optional features” on page 525, covers the following topics:
 - OnDemand Distribution Facility (ODF) on z/OS
 - Report Distribution
 - Content Manager OnDemand Toolbox
 - E-mail Notification and Delivery for Multiplatforms
- ▶ Chapter 17, “Enhancements” on page 573, covers the following topics:
 - Web Administration Client
 - Composite indexes
 - Cluster indexes
 - Cabinets
 - File name Indexing
 - LDAP security
 - 64-bit support
 - Tracing

Note: This edition is *not* a formal update of the book. We simply added the two chapters that were created during the production of the other book (*Implementing IBM Content Manager OnDemand Solutions with Case Studies*, SG24-7511). Some of the materials in these two chapters might coincide with the existing book. The original content of the book as of the May 2006, Second Edition, *has not been modified or updated* for this edition. Our purpose for the Third Edition is to make the information available to you as early as possible rather than wait until a formal update of the book.

May 2006, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

- ▶ Chapter 1, “Overview and concepts” on page 1, covers the following new topics:
 - Supported environments
 - Document access and distribution possibilities
- ▶ Chapter 2, “Administration” on page 21, covers the following new topics:
 - New function to export application
 - New function to select the font at the graphical indexer
 - New function to define the indexer parameter for Portable Document Format (PDF) data in Unicode
 - New function to do a full report browse in a folder
 - New XML batch administration tool
- ▶ Chapter 5, “Storage management” on page 105, introduces the use of IBM System Storage™ Archive Manager to be used with Tivoli Storage Manager, IBM TotalStorage® DR550 and EMC Centera.
- ▶ Chapter 8, “OnDemand Web Enablement Kit” on page 199, introduces the OnDemand Portlets.
- ▶ Chapter 9, “Data conversion” on page 249, discusses IBM AFP2WEB Services Offerings.
- ▶ Chapter 10, “Report Distribution” on page 307, is a new chapter that discusses the separately priced report distribution feature in OnDemand for Multiplatforms.
- ▶ Chapter 11, “Exits” on page 335, includes additional explanation of the following exits:

- arsufax
- arsuload
- arsuperm
- arsuprep
- arsusmxm
- arsutbl
- ▶ Chapter 12, “iSeries Common Server migration” on page 399, includes suggestions and recommendations to customers who are migrating from Spool File Archive to iSeries Common Server.
- ▶ Chapter 13, “Solution design and best practices” on page 429, is a new chapter that discusses solution design for performance and user satisfaction and includes a collection of best practices.
- ▶ Chapter 14, “Troubleshooting” on page 453, is a new chapter that includes troubleshooting FAQ, information collection, and the OnDemand Trace facility.
- ▶ Chapter 15, “Did you know” on page 479, includes the following new topics:
 - Store OnDemand
 - OnDemandToolbox
 - Date range search tip for users
 - Ad-hoc CD-ROM mastering
 - OnDemand Production Data Distribution
 - Customizing the About window
 - Modifying client behavior through the registry
 - Negative numbers in decimal fields handling
 - Message of the day
 - OnDemand bulletins

Changed information

- ▶ Chapter 2, “Administration” on page 21, contains application group deletion validation changes.
- ▶ Chapter 4, “Multiple instances” on page 77, contains parameter changes when creating an instance in OnDemand for iSeries and additional information about Linux®.
- ▶ Chapter 5, “Storage management” on page 105, includes more information about using migration policies in OnDemand for iSeries and an introduction to using the IBM System Storage Archive Manager.
- ▶ Chapter 7, “PDF indexing” on page 185, covers the discontinuation of Adobe® Acrobat® Approval.
- ▶ Chapter 8, “OnDemand Web Enablement Kit” on page 199, includes introduction updates and OnDemand Web Enablement Kit (ODWEK) deployment changes.

- ▶ Chapter 15, “Did you know” on page 479, includes more information and a new example of using the Document Audit Facility.



Overview and concepts

In this chapter, we provide an overview of the IBM DB2 Content Manager OnDemand (OnDemand) system. We describe how OnDemand manages reports and index data. We also provide information to help you better understand how OnDemand works.

In this chapter, we discuss the following topics:

- ▶ Overview of OnDemand
- ▶ Concepts
- ▶ Supported environments
- ▶ Document access and distribution possibilities

1.1 Overview of OnDemand

OnDemand supports any organization that can benefit from hard copy or microfiche replacement and instant access to information. An OnDemand system can support small office environment and large enterprise installations with hundreds of system users. It can dramatically improve productivity and customer service in many businesses by providing fast access to information stored in the system.

OnDemand processes the print output of application programs, extracts index fields from the data, stores the index information in a relational database, and stores one or more copies of the data in the system. With OnDemand, you can archive newly created and frequently accessed reports on high speed, disk storage volumes and automatically migrate them to other types of storage volumes as they age.

OnDemand fully integrates the capabilities of Advanced Function Presentation (AFP), including management of resources, indexes, and annotations. It supports full fidelity reprinting and faxing of documents to devices attached to a PC, OnDemand server, or other type of server in the network.

OnDemand provides administrators with tools to manage OnDemand servers, authorize users to access OnDemand servers and data stored in the system, and back up the database and data storage.

OnDemand gives you the ability to view documents; print, send, and fax copies of documents; and attach electronic notes to documents. OnDemand offers several advantages allowing you to:

- ▶ Easily locate data without specifying the exact report
- ▶ Retrieve the pages of the report that you require without processing the entire report
- ▶ View selected data from within a report

OnDemand provides you with an information management tool that can increase your effectiveness when working with customers. It supports the following capabilities:

- ▶ Integrates data created by application programs into an online, electronic information archive and retrieval system
- ▶ Provides controlled and reliable access to all reports of an organization
- ▶ Retrieves data that you need when you need it
- ▶ Provides a standard, intuitive client with features such as thumbnails, bookmarks, notes, and shortcuts

These features mean that OnDemand can help you quickly retrieve the specific page of a report that you require to provide fast customer service.

An OnDemand system consists of:

- ▶ Client programs and server programs that communicate over a network running the TCP/IP communications protocol
- ▶ A database manager that maintains index data and server control information
- ▶ Storage managers that maintain documents on various types of storage devices

Figure 1-1 presents an overview of the OnDemand system.

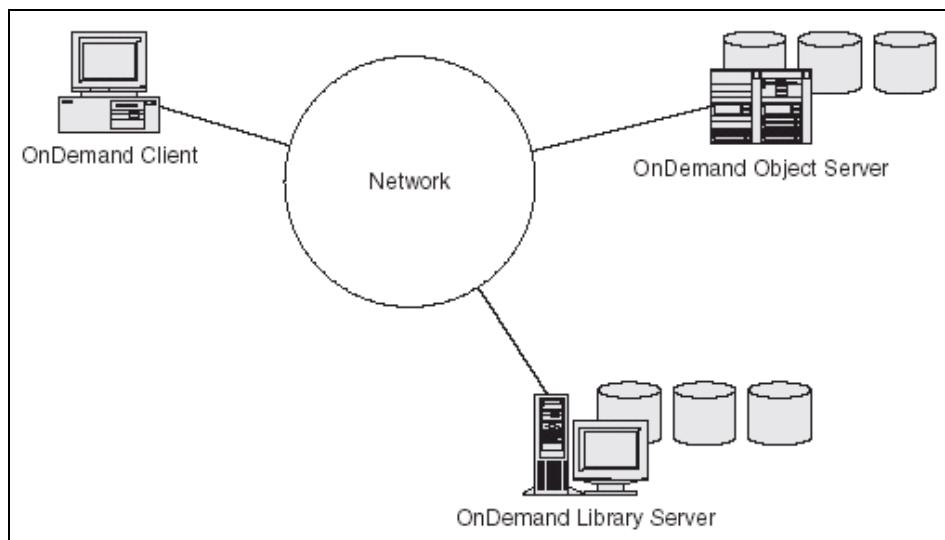


Figure 1-1 OnDemand system overview

OnDemand client programs run on PCs and terminals attached to the network and communicate with the OnDemand servers. The OnDemand library server manages a database of information about the users of the system and the reports stored on the system. The OnDemand object server manages the reports on disk, optical, and tape storage devices. An OnDemand system has one library server and one or more object servers. An object server can operate on the same server or node as the library server or on a different server or node than the library server.

OnDemand client programs operate on personal computers running on Windows. Using the client program, users can search for and retrieve reports stored on the system. Specifically, users can construct queries and search for

reports, retrieve documents from OnDemand, view, print, and fax copies or pages of documents, and attach electronic notes to pages of a document.

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

OnDemand client programs and servers communicate over a computer network supported by TCP/IP. When a user submits a query, the client program sends a search request to the 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.

1.2 Concepts

In this section, we examine some of the basic concepts of OnDemand:

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

1.2.1 Report and document

OnDemand *documents* represent indexed groups of pages. Typically an OnDemand document is a logical section of a larger *report*, such as an individual customer statement within a report of thousands of statements. An OnDemand document can also represent a portion of a larger report. For reports that do not contain logical groups of pages, such as transaction logs, 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 usually identified by date, with one or more other fields, such as customer name, customer number, or transaction number. A date is optional but highly recommended for optimizing document search performance. If there is no date field, the load ID looks similar to this example, 5179-1-0-1FAA-0-0, where the 0-0 on the end means that no date was used.

Figure 1-2 on page 5 illustrates OnDemand applications and documents. An administrator can define the BILLS application for a report that contains logical items, such as customer statements. The BILLS application uses the document

indexing method to divide the report into documents. Each statement in the report becomes a document in OnDemand. Users can retrieve a statement by specifying the date and any combination of name and number.

An administrator can define the TRANS application for a 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 OnDemand. Each group is indexed using the first and last sorted transaction values 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. OnDemand retrieves the group that contains the value entered by the user.

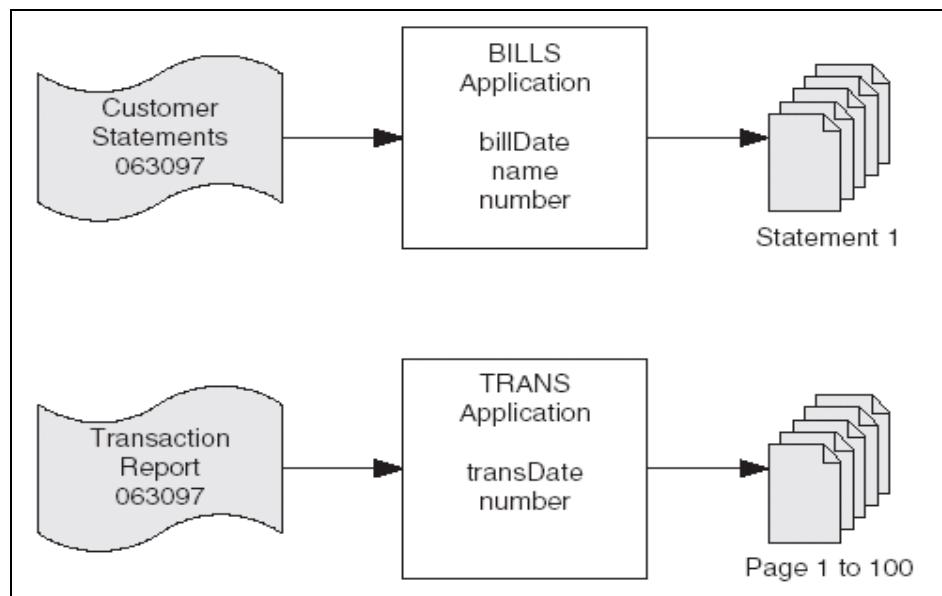


Figure 1-2 Applications and documents

1.2.2 Application, application group, and folder

The terms *application*, *application group*, and *folder* represent how OnDemand stores, manages, retrieves, views, and prints reports and index data. When defining a new report or type of data to OnDemand, an administrator must create an application and assign the application to an application group.

Note: The administrator must first create an application group if one does not exist.

Before users can search for and retrieve documents, an administrator must create or update a folder to use the application group and application.

Application

An *application* describes the physical characteristics of a report to OnDemand. Typically you define an application for each program that produces output to be stored in 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 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 OnDemand. When you load a report into OnDemand, you must identify the application group where OnDemand will load the index data and store the documents.

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

Folder

A *folder* is the user's way to query and retrieve data stored in OnDemand. A folder provides users with a convenient way to find related information stored in 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 screen 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 called "Student Information" might contain transcripts, bills, and grades, which represent information stored in different application groups, defined in different applications, and created by different programs.

Figure 1-3 illustrates the concepts described in this section.

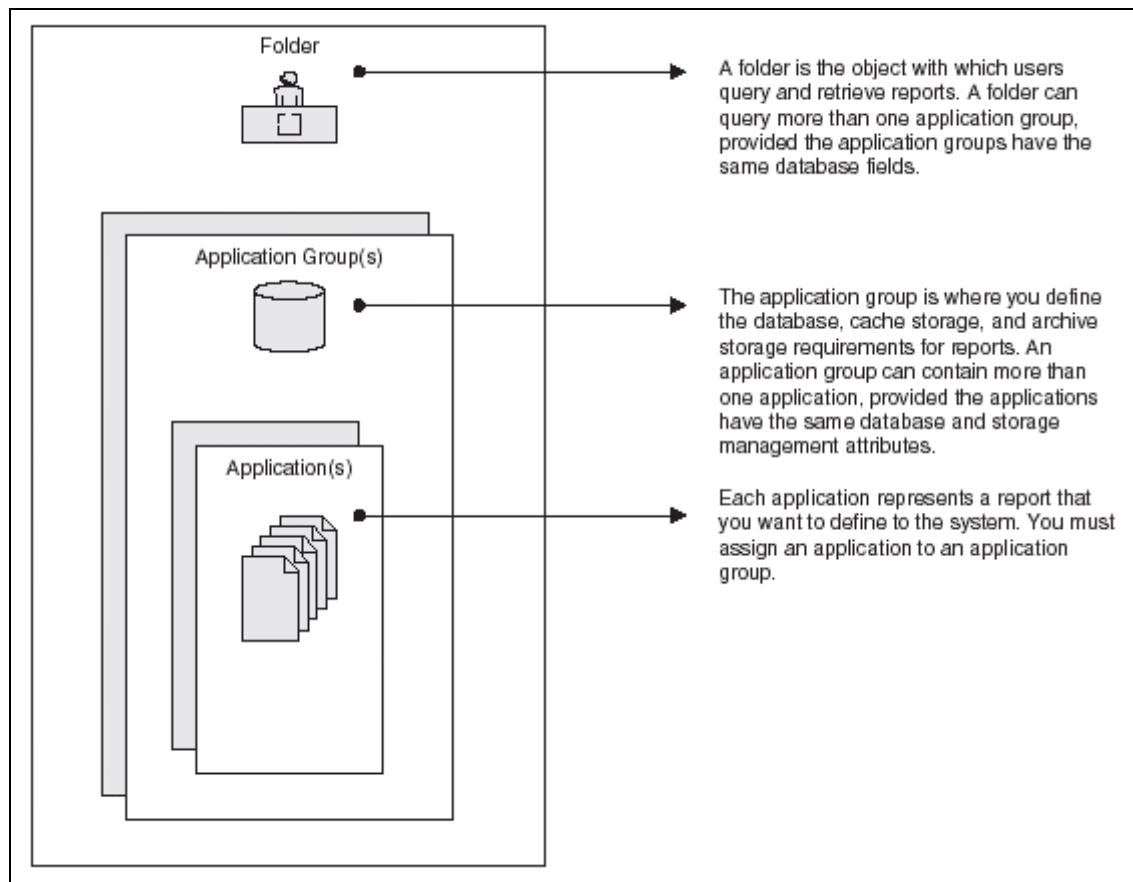


Figure 1-3 The concepts of folders, application groups, and applications

Figure 1-4 shows some examples for the described concepts.

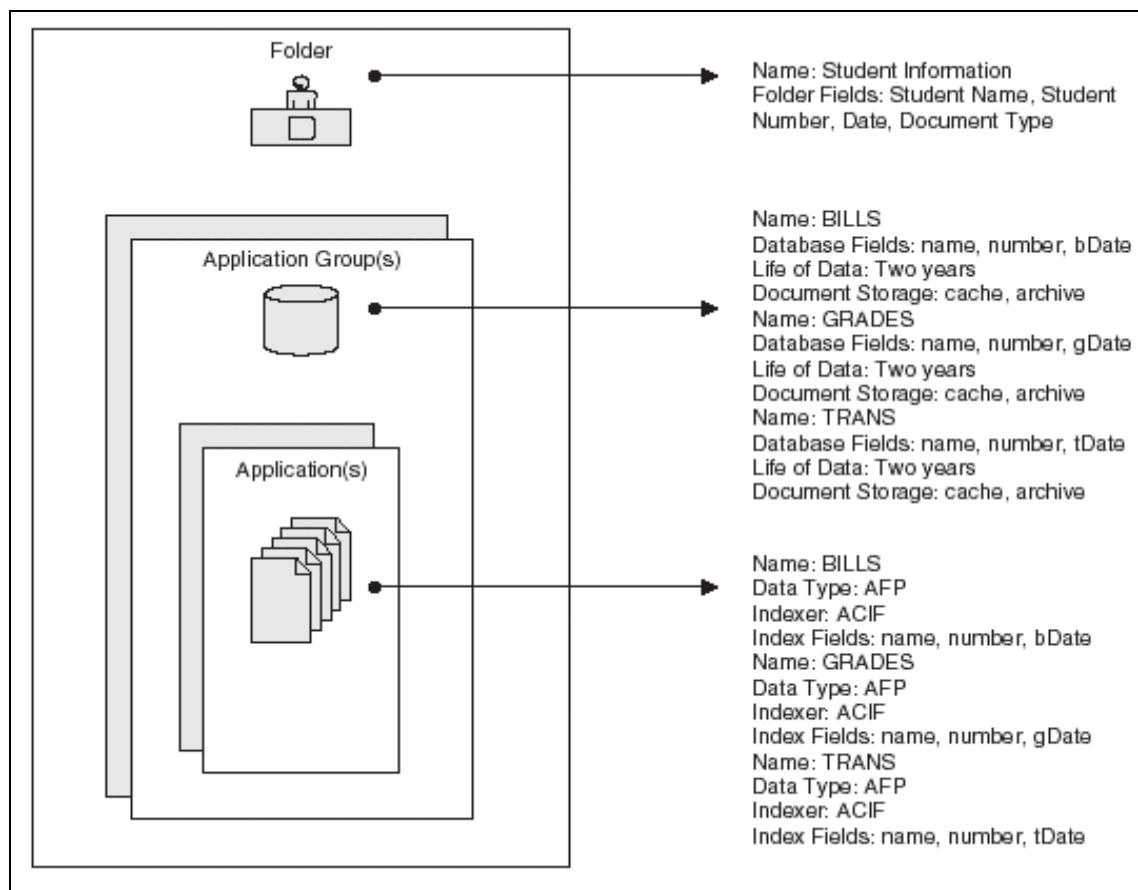


Figure 1-4 Examples of folders, application groups, and applications

1.2.3 Indexing methods

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 policies, and statements. Each of the items in a report can be individually indexed on values such as account number, customer name, and balance. OnDemand supports up to 32 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 OnDemand.

Report indexing

Report indexing is used for reports that contain many pages of the same kind 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. OnDemand stores the report as groups of pages and indexes each group.

When reports include a sorted transaction value (for example, invoice number), OnDemand can index the data on the transaction value. This 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.

1.2.4 OnDemand server and its components

The OnDemand server environment includes a library server and one or more object servers that reside on one or more nodes connected to a TCP/IP network.

Library server and object server

An OnDemand *library server* maintains a central database about the reports stored in OnDemand. The database also contains information about the objects defined to the system, such as users, groups, printers, application groups, applications, folders, 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.

An OnDemand *object server* maintains documents on cache storage volumes and, optionally, works with an archive storage manager to maintain documents on archive media, such as optical and tape storage libraries. An object server loads data, retrieves documents, and expires documents. The major functions that run on an object server are the cache storage manager, OnDemand data loading and maintenance programs, and optionally, the archive storage manager.

The basic 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 archive storage manager to the single library or object server configuration, to maintain documents on archive media.

You can also configure your OnDemand system with a library server on one node and one or more object servers on different nodes. This configuration is known as a distributed library/object server system. The distributed library or object server configuration supports caching of documents on different servers. You can add an archive storage manager to one or more of the object servers to maintain documents on archive media attached to different servers.

OnDemand server component

An OnDemand server environment contains several components:

- ▶ A *request manager* provides client, network, and operating system services, security, and accounting. The request manager resides 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 DB2. The database manager resides on the library server.
- ▶ Database *control information* is information about the users, groups, application groups, applications, folders, 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 resides on the library server.
- ▶ A *cache storage manager* maintains documents in cache storage. Cache storage is for high-speed access to the most frequently used documents.
- ▶ An *archive storage manager* is an optional part of the system. The archive storage manager is for the long-term storage of one or more copies of documents on archive media, such as optical and tape storage libraries.
- ▶ A *download facility* automatically transfers spool files to a server at high speed. We recommend that you use Download for OS/390®, a licensed feature of Print Services Facility™ (PSF) for OS/390. Download provides the automatic, high-speed download of JES spool files from an OS/390 system to OnDemand servers. The download facility is not applicable to the iSeries server.
- ▶ *Data indexing and conversion programs* can create index data, collect required resources, and optionally convert line data reports to AFP data. OnDemand provides several indexing programs:
 - The *AFP Conversion and Indexing Facility (ACIF)* can be used to index S/390® line data, ASCII data, and AFP files, collect resources necessary to view the reports, and convert line data files to AFP data.
 - The *OS/400® Indexer* can be used to index a variety of data types and is the most common OnDemand indexer for OS/400 spooled files.
 - The *OnDemand PDF Indexer* can be used to create index data for Adobe Acrobat Portable Document File (PDF) files. The OnDemand Generic

Indexer can be used to create index data for almost any other type of data such as HTML documents, Lotus® WordPro documents, TIFF files.

- ▶ *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 OnDemand server.
- ▶ *Archived reports and resources.*
- ▶ A *server print* facility allows users to reprint a large volume of documents at high speed. OnDemand uses Infoprint, which must be purchased separately, to manage the server print devices.
- ▶ OnDemand *management programs* maintain the 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.

The following sections provide additional information for:

- ▶ The OnDemand request manager
- ▶ The OnDemand database manager
- ▶ The OnDemand storage manager
- ▶ Download facility
- ▶ Data indexing and loading
- ▶ OnDemand management programs

Request manager

The request manager processes search requests from OnDemand client programs. When a user enters a query, the client program sends a request over the network to the request manager. The request manager works with the database manager to compile a list of the items that match the query and returns the list to the client program. When the user selects an item for viewing, the request manager sends a retrieval request to the cache storage manager, if the document resides in cache storage, or to the archive storage manager, if the document resides in archive storage. The storage manager retrieves the document and, optionally, the resources associated with the item. The OnDemand client program decompresses and displays the document.

OnDemand management programs include utilities that maintain the database and cache storage, including the ability to automatically migrate data from the database and cache storage to archive storage. These programs use the services of the request manager to manage index data, documents, and resource files.

When a user logs on to the system, OnDemand assigns a unique transaction number to that instance of the client program. All activity associated with that instance of the client program contains the same transaction number. The request manager records messages generated by the various OnDemand programs in the system log, for example, logon, query, and print. The messages contain the transaction number, user ID, time stamp, and other information. Administrators can open the system log folder and view the messages.

OnDemand also provides a system log user exit so that you can run a user-defined program to process messages. For example, you can design a user-defined program to send an alert to an administrator when certain messages appear in the system log. The messages in the system log can also be used to generate usage and billing reports.

Database manager

OnDemand uses a database management product, such as DB2, to maintain the index data for the reports that you load into the system. The database manager also maintains the OnDemand system tables that describe the applications, application groups, storage sets, folders, groups, users, and printers that you define to the system. You should periodically collect statistics on the tables in the database to optimize the operation of the OnDemand database.

Storage manager

The OnDemand *cache storage manager* maintains a copy of documents, usually temporarily, on disk. The cache storage manager uses a list of file systems to determine the devices available to store and maintain documents. You typically define a set of cache storage devices on each object server so that the data loaded on the server can be placed on the fastest devices to provide the most benefit to your users.

For multiplatforms and z/OS, the cache storage manager uses the **arsmain** command to migrate documents from cache storage to archive media and to remove documents that have passed their life of data period. For iSeries, the Start Disk Storage Management (STRDSMOND) command starts the Disk Storage Management task that manages data from between disk and the archive storage manager.

OnDemand also supports an *archive storage manager*, such as Tivoli Storage Manager for Multiplatforms, object access method (OAM) for z/OS, Virtual Storage Access Method (VSAM) for z/OS, and Archive Storage Manager for iSeries. The archive storage manager maintains one or more copies of documents on archive media, such as optical or tape storage libraries. You decide the types of archive media that your OnDemand system must support, configure the storage devices on the system, and define the storage devices to

the archive storage manager. To store application group data on archive media, you must assign the application group to a storage set that identifies a storage node that is managed by the archive storage manager.

Download facility

The download facility is a licensed feature of PSF for OS/390. It provides the automatic, high-speed download of JES spool files from an OS/390 system to an OnDemand for Multiplatforms server. You can use the download facility to transfer reports created on OS/390 systems to the server, where you can configure OnDemand to automatically index the reports and store the reports and index data on the system. The download facility operates as a JES functional subsystem application (FSA) and can automatically route jobs based on a JES class or destination, reducing the need to modify JCL. The download facility uses TCP/IP protocols to stream data at high speed over a LAN or channel connection from an OS/390 system to the OnDemand server.

Data indexing and loading

The reports that you store in OnDemand must be indexed. OnDemand supports several types of index data and indexing programs. For example, you can use ACIF to extract index data from the reports that you want to store on the system. An administrator defines the index fields and other processing parameters that ACIF uses to locate and extract index information from reports.

OnDemand data loading programs read the index data generated by ACIF and load it into the OnDemand database. The data loading programs obtain other processing parameters from the OnDemand database, such as parameters used to segment, compress, and store report data in cache storage and on archive media. If you plan to index reports on an OnDemand server, you can define the parameters with the administrative client. The administrative client includes a Report Wizard that lets you create ACIF indexing parameters by visually marking sample report data. OnDemand also provides indexing programs that can be used to generate index data for Adobe Acrobat PDF files and other types of source data, such as TIFF files.

For OS/400, the OS/400 Indexer can index a variety of data types for OS/400 spooled files. Refer to the following publications for details about the indexing programs provided with OnDemand for various platforms:

- ▶ *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235
- ▶ *IBM Content Manager OnDemand for iSeries Common Server - Indexing Reference*, SC27-1160
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - Indexing Reference*, SC27-1375

Figure 1-5 shows an overview of the data indexing and loading process.

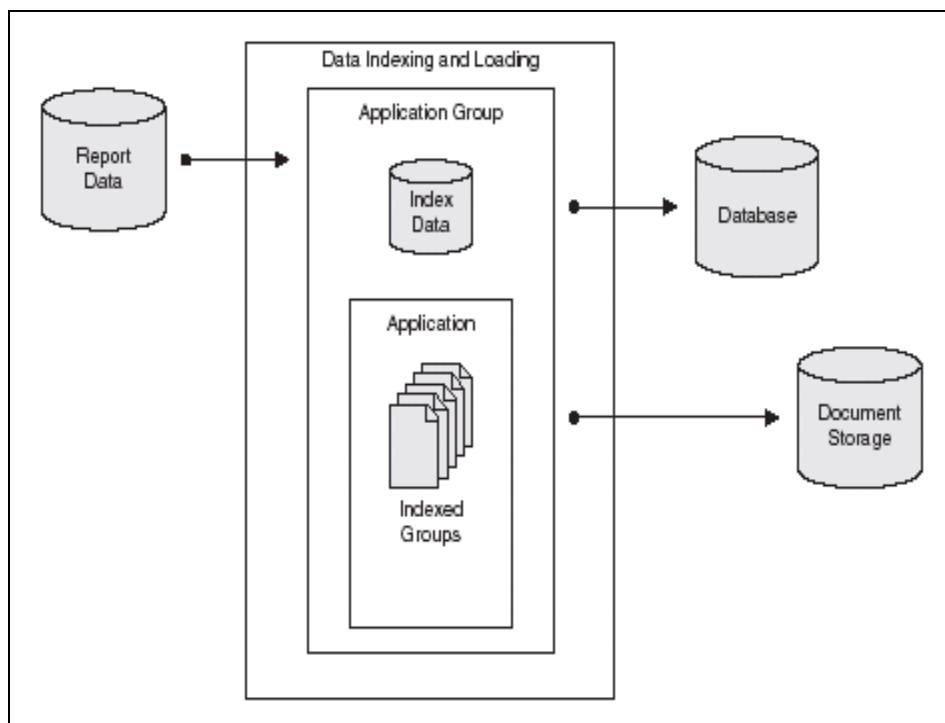


Figure 1-5 Data indexing and loading process

The OnDemand data loading program first determines whether the report needs to be indexed. If the report needs indexing, the data loading program calls the appropriate indexing program. The indexing program uses the indexing parameters from the OnDemand application to process the report data. The indexing program can extract and generate index data, divide the report into indexed groups, and collect the resources needed to view and reprint the report.

After indexing the report, the data loading program processes the index data, the indexed groups, and the resources using other parameters from the application and application group. The data loading program works with the database manager to update the OnDemand database with index data extracted from the report. Depending on the storage management attributes of the application group, the data loading program might work with the cache storage manager to segment, compress, and copy report data to cache storage and the archive storage manager to copy report data to archive storage.

Management programs

OnDemand provides programs to maintain and optimize the database and maintain documents on cache storage. An administrator usually determines the processing parameters for these programs, including the frequency with which the programs should run. When you create an application group, you specify other parameters that these programs use to maintain the report data stored in the application group.

For example, when creating an application group, the administrator specifies how long documents should be maintained on the system and whether index data should be migrated from the database to archive media. The programs use the information to migrate documents from cache storage to archive media, delete documents from cache storage, migrate index data from the database to archive media, and delete index data from the database. These functions are useful because OnDemand can reclaim the database and cache storage space released by expired and migrated data. We recommend that you configure your OnDemand system to automatically start these management programs on a regular schedule, usually once every night or week.

The archive storage manager deletes data from archive media when it reaches its storage expiration date. An administrator defines management information to the archive storage manager to support the OnDemand data it manages. The management information includes the storage libraries and storage volumes that can contain OnDemand data, the number of copies of a report to maintain, and the amount of time to keep data in the archive management system.

OnDemand and the archive storage manager delete data independently of each other. Each uses its own criteria to determine when to remove documents. Each also uses its own utilities and schedules to remove documents. However, for final removal of documents from the system, you should specify the same criteria to OnDemand and the archive storage manager.

1.3 Supported environments

OnDemand can be installed on a variety of platforms. Three products are available based on platform:

- ▶ OnDemand for Multiplatforms
- ▶ OnDemand for z/OS
- ▶ OnDemand for iSeries

OnDemand for Multiplatforms

OnDemand for Multiplatforms supports diverse IBM and non-IBM platforms and diverse operating systems. You can install OnDemand as a stand-alone product on one platform or within a global archive network.

Multiplatforms that OnDemand can run include:

- ▶ AIX V5.1 or later on IBM eServer p5 or later
- ▶ XP-UX version 11i or later on HP 9000 Server rp2400 Series or later
- ▶ Sun™ Solaris™ Version 8 on Sun Fire™ B100s Blade Server
- ▶ Microsoft Windows 2000, 2003 Server
- ▶ Linux on Intel®-based 1 GHz or greater processor
 - Red Hat Enterprise Linux (RHEL) AS or ES
 - SUSE LINUX Enterprise Server (SLES) 8 SP 3

Linux support is new in Content Manager V8.3. With the addition of this support, you can choose a cost-effective platform, while fully leveraging important information across daily business operations.

Note: Because many Linux versions are available, make sure to use only the supported versions for OnDemand. If you use a nonsupported version, you might run into problems.

- ▶ Linux for zSeries® on any IBM eServer zSeries model that supports running Linux
 - Red Hat Enterprise Linux AS or ES
 - SUSE LINUX Enterprise Server 8 SP 3

Note: Neither OnDemand PDF indexer nor OnDemand Report Distribution is supported under Linux for zSeries.

OnDemand for z/OS

Content Manager OnDemand for z/OS is high performance middleware that capitalizes on strategic IBM hardware and software to provide a highly reliable system for enterprise report management and electronic statement presentation. It works on any zSeries with OS/390 Version 2 Release 8 or later, operational TCP/IP and UNIX® System Services, and DB2 for OS/390 Version 6 or later.

Content Manager OnDemand for z/OS includes the capability to mix and match servers to support the architecture and geographic needs of the customers, while allowing the data to be kept closer to the requesting users, minimizing network traffic. For example, a Chicago-based company with an z/OS-based library

server and object server can add an additional object server in Dallas that can be on any OnDemand platform. Or you can choose to run OnDemand library server on AIX or Microsoft Windows NT®, with the object servers on z/OS.

OnDemand for iSeries

Content Manager OnDemand for iSeries is high performance middleware that capitalizes on IBM hardware. With OnDemand, the iSeries or AS/400 becomes an archive server for OS/400 business applications or applications on other host systems. It works on any iSeries with OS/400 V5 or later.

Note: The library server and the object server must be on the same iSeries partition. It is not possible to mix servers as is possible with OnDemand for Multiplatforms and OnDemand for z/OS.

1.4 Document access and distribution possibilities

Users' needs vary. They can be different from one company to another. They might even vary inside of the same company. Users can also be the customers or partners of a company. They might have access to a network or require some documents to be available without any network connection. They might need advanced functions such as graphical annotations or audit facilities.

Users might have to search for documents when a customer calls in for a particular transaction. They might need to view special monthly reports after the reports are generated. They might access information of different types and from different sources, such as unstructured documents stored either in OnDemand or in some third-party offerings and structured information stored either in DB2 or in third-party database offerings.

OnDemand offers a large choice of solutions that cover many of the users' needs. In this section, we introduce:

- ▶ Document access methods for OnDemand
- ▶ Document distribution possibility using OnDemand
- ▶ Federated search: WebSphere® Information Integrator Content Edition

1.4.1 Document access

Users can access archived documents from OnDemand using one of the following two methods:

- ▶ Connected to the OnDemand server mode
- ▶ Disconnected mode

Connected to the OnDemand server mode

A network connection is available between the OnDemand client used by the user and the OnDemand server. The user is able to access all the archived documents including the most recent archives, based on the user's security rights.

The server might be any of the OnDemand server types. The client can be the OnDemand client, a browser client, or a custom client.

Disconnected mode

Data is extracted from an OnDemand server and is written to a media that can be easily distributed. Users access the OnDemand data from the media in the same way that they access data that is stored in an OnDemand server. For example, you can store all your invoices of a period in a media and loaded it to a mobile computer. Sales representatives can then access the customer invoices on their mobile computers without any network connections while they are in the field.

There are two data distribution options:

- ▶ CD-ROM - Client Data Distribution (also known as *ad-hoc CD-ROM*)
This option is designed for low-volume, ad-hoc building of a CD-ROM by a user with the OnDemand client. Refer to 15.8, “Ad-hoc CD-ROM mastering” on page 502, for more information.
- ▶ CD-ROM - Production Data Distribution services offering
This option is designed for high-volume, batch processing of input files and documents and the production of multiple copies of CD-ROMs. Refer to 15.9, “OnDemand Production Data Distribution” on page 511, for more information.

1.4.2 Document distribution possibility

Users access documents whenever they need them through one of the OnDemand clients. The documents can also be distributed to users whenever the documents are available.

OnDemand offers the *Report Distribution* optional feature, which allows documents to be distributed and shared via e-mail. Report Distribution automatically groups reports and portions of the related reports together, organizes them, and creates bundles that can be sent through e-mail to multiple users or sent to the customer's printing environment for printing.

Refer to Chapter 10, “Report Distribution” on page 307, for more information.

1.4.3 Federated search using WebSphere Information Integrator Content Edition

You might require access to information stored in OnDemand as well as other repositories that contain structured and unstructured data, within and outside of your company. You can conduct a federated search using *WebSphere Information Integrator Content Edition*. Although this IBM Redbooks publication covers OnDemand, the product, it might be useful for you to know that you can use WebSphere Information Integrator Content Edition to perform a federated search against content in different repositories including the OnDemand repository.

WebSphere Information Integrator Content Edition V8.3 enables you to access and work with a wide range of information sources. It is a single and bi-directional interface that enables multiple disparate content sources to look and act as one system. This flexible and highly scalable abstraction layer makes applications repository independent.

WebSphere Information Integrator Content Edition provides the following functions and services:

- ▶ Access to the underlying content and workflow systems
- ▶ Check in, check out, and modify content
- ▶ View and update metadata, security, and annotations
- ▶ Create and work with renditions, compound documents, workflow tasks, and queues
- ▶ View major standard business document formats, including TIFF and MODCA
- ▶ Monitor content, content folders, workflow items, and queues for changes, and trigger actions when a change occurs
- ▶ Full read-write function
- ▶ Ability to organize and work with content assets and workflow items as though they are managed in one system
- ▶ Metadata mapping, federated search, and single sign-on
- ▶ A uniform super-set APIs to eliminate programming using multiple APIs from different vendors
- ▶ Real-time content views of content and workflow accessed in place remove the need to access each repository individually

WebSphere Information Integrator Content Edition provides many technology benefits:

- ▶ Service-oriented architecture
- ▶ Fully J2EE™-compliant and Web services compatible

- ▶ Support WebSphere Application Server and BEA WebLogic Server
- ▶ Support component distribution and load balancing
- ▶ Provide SOAP interface for Web services applications
- ▶ Support URL-addressable functions

To access data in a different content repository, WebSphere Information Integrator Content Edition provides out-of-box connectors that are available for the following products and systems:

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

WebSphere Information Integrator Content Edition also provides a toolkit for you to develop, configure, and deploy content connectors for additional commercial and proprietary repositories. Sample connectors are provided.

For more information about WebSphere Information Integrator Content Edition, refer to the following Web address:

http://www.ibm.com/software/data/integration/db2ii/editions_content.html



Administration

An extremely important aspect of an OnDemand system is the effective design and implementation of a strategy regarding system administration from a report administration perspective and from a user authority and responsibility perspective. The focus of this strategy should be to ensure that the system is planned in a manner that provides the greatest functionality and the best performance as the system matures.

In this chapter, we discuss the following topics:

- ▶ Report administration
- ▶ User and group administration

2.1 Report administration

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

The system components that are required for creating, retrieving, and viewing an OnDemand report are a storage set, an application group, an application, and a folder. These elements, in combination, allow the OnDemand administrator to define and create a report definition that can then be used to index and load data into OnDemand. Figure 2-1 illustrates the relationship of these elements in a typical OnDemand system.

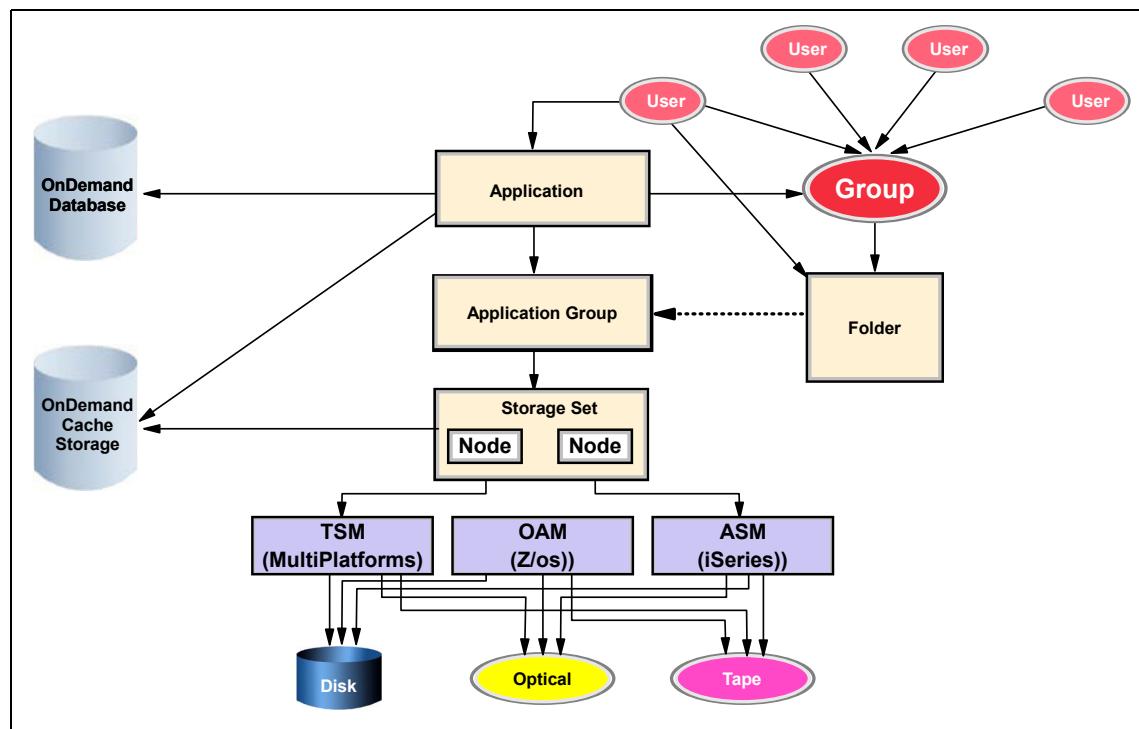


Figure 2-1 OnDemand system components relationship

2.1.1 Storage sets

A *storage set* contains one or more storage nodes that can be used by several application groups that have the same archive storage 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 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 required.

For a more in-depth look into storage management, see Chapter 5, “Storage management” on page 105. Excerpts from that chapter are repeated here to introduce the various report administration related topics.

2.1.2 Application groups

An *application group* is 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 defines the data that is to be loaded into the database. In the following sections, we look closer at the aspects of an application group definition that can contribute to a successful OnDemand system implementation.

Database information

The database information section of the application group definition process (Figure 2-2) requires that decisions be made concerning the number of rows to be stored in each database table and the number of report loads to be included in each database table. These values are important to system performance and maintenance.

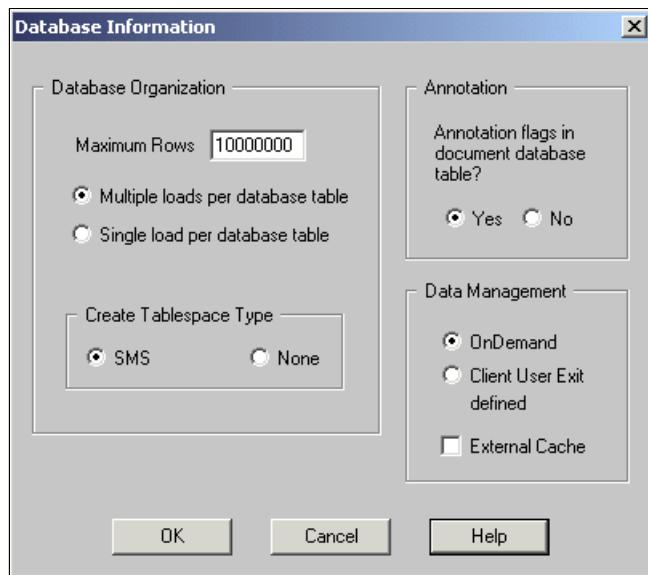


Figure 2-2 Database information

Maximum rows

The *maximum rows* value, which determines how many data rows will be loaded into each database table, is used for segmenting the index data and determining when to close a database table and open a new one. We recommend that you use the default value of 10000000 rows for balancing the performance of data loads and queries. The number of rows specified should be large enough to handle the largest possible input report file. You should decrease the value if there is a small amount of data associated with the application group, thereby increasing query performance without adversely affecting data load performance.

Loads per database table

The number of loads per database table can be set to multiple or single. If *multiple loads* is chosen, every time that a report is loaded into an application group, the index records are added to the database table that is currently open for the application group. When the current application group table reaches the maximum rows value, the table is closed and a new table is opened. We recommend that you use multiple loads per database table.

If *single load* is chosen, a new database table is used for each load of a report into the application group. The maximum rows value is used to calculate the space allocation for the single load tables. However, a single load per database table is no longer supported. Existing application groups with the options selected can still be used, but new application groups cannot use this option.

Storage Management

The storage management settings (Figure 2-3) determine how long report data and indexes will be kept in cache storage before being expired. There are also options that determine how soon data will be migrated to archive storage after the report load is completed.

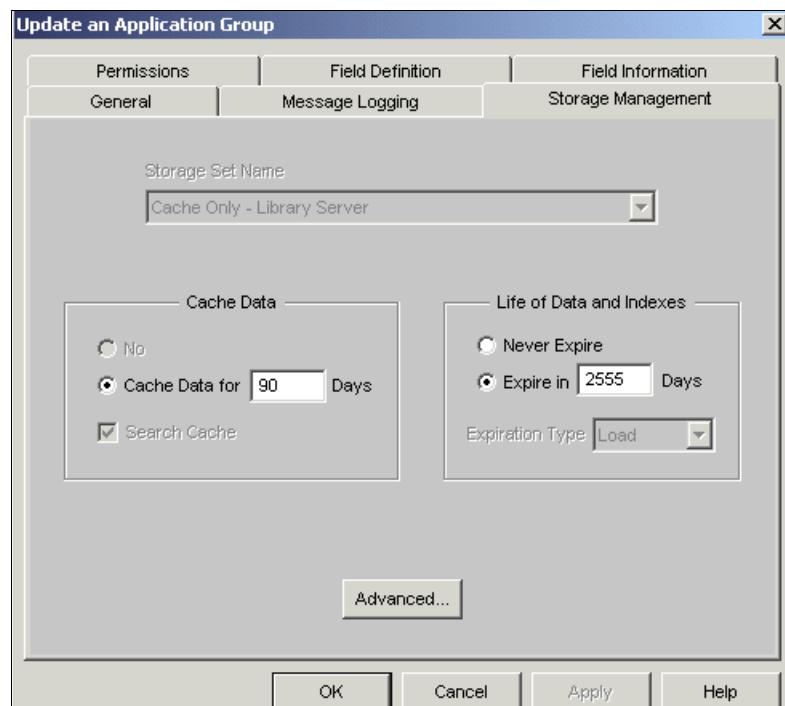


Figure 2-3 Application group storage management

Cache Data

The Cache Data setting determines if the report data will be stored in disk cache, and if so, how long it will be kept in cache before it is expired. If the Cache Data for *nn* Days parameter is selected, then Search Cache is always selected.

Search Cache determines whether OnDemand searches cache storage when users retrieve documents from the application group. When you set the cache data to No, you can configure 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, a storage set that supports archive storage must be selected.

Note: Data that is retrieved often should generally remain in cache until it is no longer needed by 90% of 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 OnDemand system before they are deleted from an 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 has been reached, the expiration processing can be used to expire the data from the system.

Expiration Type

The Expiration Type determines how report data, indexes, and resources are expired. 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, determine when OnDemand will delete the data. Data that has been stored in archive storage is deleted by the storage manager based on the archive expiration date. We recommend that you set Expiration Type to *Load*.

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 have been reached. 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 of time, and the data will not expire for the period.

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 if the expiration data has been reached, which might result in long processing times.

When the **arsmain** expiration process is run, data is only deleted from the application group if the upper threshold for the size of cache storage has been reached. By default, the cache threshold is 80 percent. A lower threshold can be forced by the expiration command parameters. However, unless there is some reason that cache needs to be cleared, leaving data in cache will improve retrieval performance.

The iSeries server does not use a cache upper threshold. If the cache data for days duration has passed and Disk Storage Manager is run, then the data is deleted from cache immediately.

Field Information

The Field Information tab (Figure 2-4) is used to define the attributes of the database fields to make up the 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.

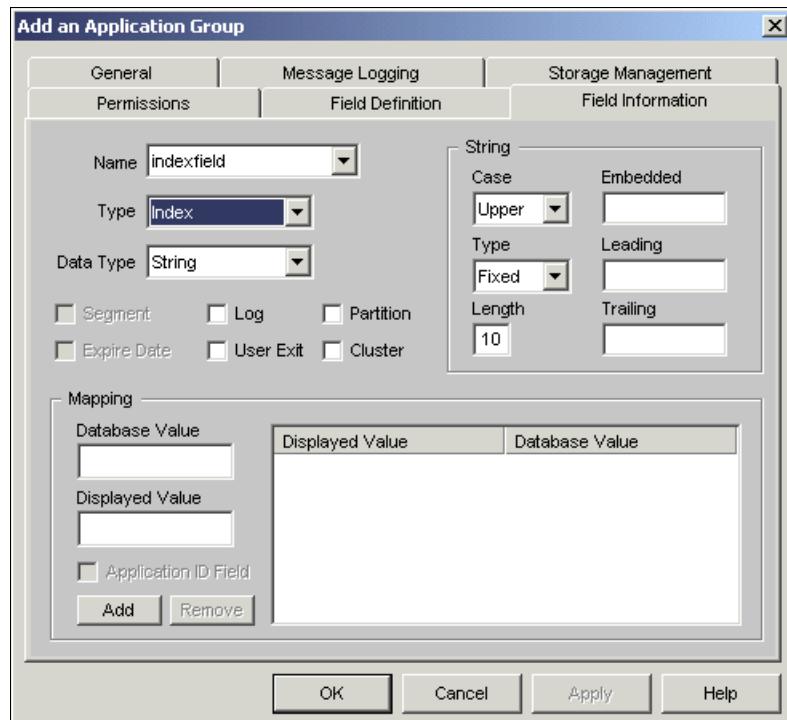


Figure 2-4 Application group Field Information tab

If multiple applications will be part of the application group, select the Application ID Field to uniquely identify each application in an application group. If it is possible that more than one application will be part of an application group, you should select the Application ID Field.

Note: Be sure that all of the database fields needed are included *before* the application group is added to the OnDemand system. Database fields *cannot* be added *after* the application group has been created.

Next we consider the following Field Information attributes in detail:

- ▶ Type
- ▶ Segment
- ▶ Application ID Field

Type

The Type attribute determines the manner in which the database field is used by OnDemand. There are three main types of attributes: *Index*, *Filter*, and *Not in database*.

A field should have a type of *index* if it is used to uniquely identify a document or if it is frequently used when searching for documents in the application group. Designating a field as an index serves to enhance query performance but increases required overhead 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 that are to be included in the document hit list.

A field should have a type of *filter* if it does not uniquely identify a document of the file and that it is usually used in conjunction with an index field during folder queries.

Important: Folder queries using filter fields alone result in a sequential scan through database tables. An index field should always be included in folder queries. For more information about folders, see 2.1.4, “Folders” on page 35.

A field should have a type of *not in database* if the field contains the same data value for every document in the input data. A value for that field will be stored in the segment table pointing to the database index records rather than storing the same value over and over again in each row of the database.

A thorough understanding of the way that users will search for documents in the system is required before making decisions about which fields should be indexes and which fields should be filters. Only fields that are going to be heavily used when searching for and retrieving documents should have a type of index. An index field should always be included in a folder query.

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. If the application group is defined for multiple loads per database table, we highly recommend that you define a segment date for the application group. By using a segment date to limit folder queries to a single table or a limited set of tables, performance is significantly improved. The segment date is especially important for application groups that contain a large amount of data.

Note: The date field that is used for the segment date should 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 overhead.

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.

Password validation now required for deletion

When an application group is deleted, not only is the system definition gone, the data that was loaded in the application group is deleted as well. To provide another level of protection against the accidental deletion of an application group, a password is now required for this action.

The password window is prompted with the user ID that is currently being used. The server then validates the password and deletes the application group only after the password validation is successful. The user is then prompted with the Delete Application Group window, followed by the Confirm Delete Application Groups window. Selecting OK causes the application group to be deleted. For more details, refer to Technote #1213631 at the following Web address:

<http://www.ibm.com/support/docview.wss?uid=swg21213631>

2.1.3 Applications

An application defines the data that is to be indexed and loaded, associates the data with an application group, and specifies the type of indexing process to be performed on the data. It also defines any logical views to be put in place for the end users and determines any special print options to be used with the data. In this section, we consider some of the load information attributes.

Load Information

Load Information specifies the processing and resource information that the OnDemand loader uses to load the input data onto storage volumes and to load the associated index data into the OnDemand database. The File Format, Preprocessor Parameters, and Postprocessor Parameter (Figure 2-5 on page 31) are defined as part of load information:

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

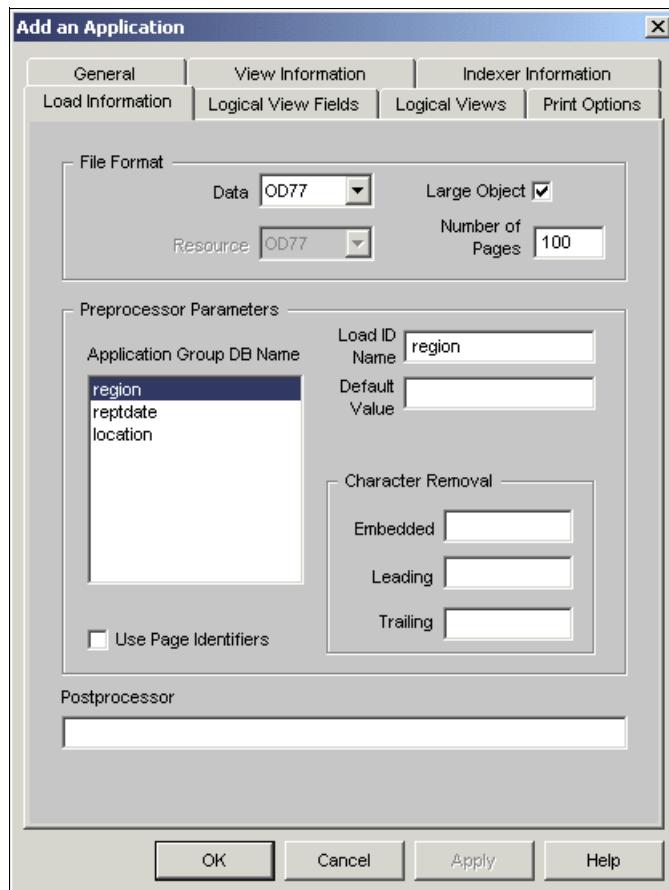


Figure 2-5 Application load information

Large Object support

In the File Format section, you can set to support large objects. Large Object support is used to improve load and retrieve performance by dividing the document into smaller parts for loading and creating index information based on this document segmentation. Documents are retrieved faster due to the smaller segment sizes that are sent across the network.

When a document is retrieved for viewing, only the first part of the document is returned from the server to the client. Additional parts of the document are sent from the server to the client as the user moves to different pages in the document. Advanced Function Presentation (AFP) Conversion and Indexing Facility (ACIF) and OS/400 are the two indexers that can be used to enable large object support. Invoking Large Object support generates an INDEXOBJ=ALL entry

in the indexing parameters that enables the generation of large object indexing information.

When Large Object is selected, the number of pages parameter must also be specified. Number of pages determines how many pages will be included by OnDemand in each large object segment.

If Large Object is not selected, the compressed object size parameter is included in the load information. The compressed object size specifies the size in kilobytes of each stored block of data. We recommend that you use the default size of 100 KB blocks.

Exporting an application

It is not possible to export an application to application groups that have different database fields or attributes. However, it is possible to export applications to a different server as long as 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).

You can use the following methods to export applications:

- ▶ Select the **Export** toolbar button.
- ▶ Right-click and select the **Export** menu item.
- ▶ Drag the objects from the source server to the target server in the left pane of the Administrative client main window.

If an application identifier is used in the application group, remember to add it into the target application group; otherwise the export fails. Likewise, there should not be an existing application that has the same application identifier in the target application group. For more information, refer to Technote #1178782, which you can find at the following Web address:

<http://www.ibm.com/support/docview.wss?uid=swg21178782>

Selecting font by line data graphical indexer

With the new version of IBM DB2 Content Manager OnDemand for Multiplatforms, the font that is used by the line data graphical indexer to display a document can now be changed from within the line data graphical indexer at the OnDemand Administrative Client.

To change the font:

1. Right-click the document and select **Font → Select...**
2. In the Font dialog box, select a font, font style, and size. Click **OK**.

The document refreshes using the new font choice.

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

To change the font setting back to the default font, right-click the document and select **Font → Reset**.

Note: If the font is changed while using the administration 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.

You can find more detail in Technote #1215957, which is available at the following Web address:

<http://www.ibm.com/support/docview.wss?uid=swg21215957>

Defining an indexer parameter for PDF data in Unicode format

IBM DB2 Content Manager OnDemand for Multiplatforms has added support for Portable Document Format (PDF) data in Unicode format when using the PDF graphical indexer or the PDF indexer. To define indexer parameters using Unicode data, a new check box called *Output Hexadecimal Strings* has been added to the Indexer Properties window (Figure 2-6 on page 34).

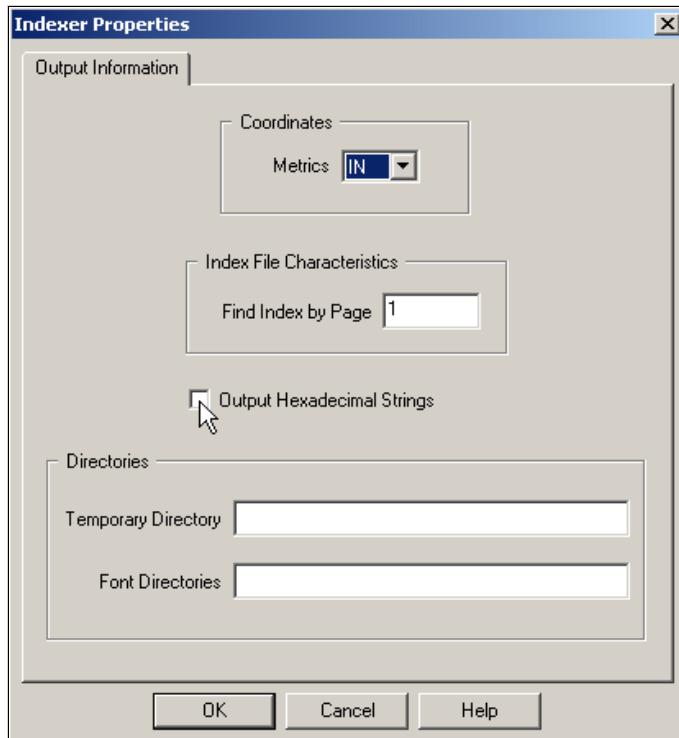


Figure 2-6 Indexer Properties for PDF graphical Indexer

You must select the Output Hexadecimal Strings check box prior to defining trigger or field parameters. After you select the check box, click **OK** to confirm the change. When a trigger or field is defined, the selected text is displayed as a hexadecimal string in the window.

The attribute value of the trigger parameter is also represented as a hexadecimal string. In Example 2-1, the trigger parameter contains an attribute value in hexadecimal format instead of a text string. The hexadecimal string is highlighted in bold.

Example 2-1 Trigger parameter with hexadecimal string

```
TRIGGER1=UL(0.75,0.84),LR(1.52,1.46),*,X'5041594D454E54'
```

For field parameters, the default value and constant value are in the hexadecimal format instead of a text string as well. In Example 2-2, the field parameters for the default field and constant field are both in hexadecimal strings. The hexadecimal strings are shown in bold text.

Example 2-2 Field parameters with hexadecimal strings

```
FIELD1=UL(0.82,1.96),LR(4.10,2.32),0,(TRIGGER=1,BASE=0,DEFAULT=X'616264')
FIELD2=X'414D4F554E54'
```

For more information, refer to Technote #1219572, which you can find at the following Web address:

<http://www.ibm.com/support/docview.wss?uid=swg21219572>

2.1.4 Folders

A *folder* is the interface that allows a user to search for reports and documents that have been stored in the OnDemand system. The user enters index search criteria for an application group into the folder search fields and a document hit list is constructed based on the results of the query. The folder can be customized to provide the look and feel that is desired for the users of the OnDemand system. The folder definition process allows the OnDemand administrator to grant specific permissions for users of the folders.

In this section, we consider several folder parameter settings (Figure 2-7 on page 36) that can impact document retrieval performance. We also discuss a new functionality called the Full Report Browse that has been added to the OnDemand Administrative Client.

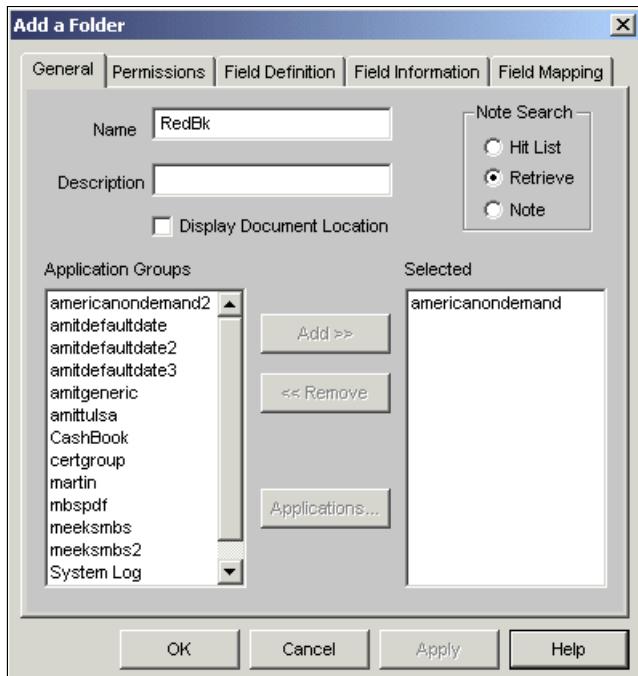


Figure 2-7 Folder general information

Display Document Location

The Display Document Location setting (Figure 2-7) causes OnDemand to display an icon next to each entry in a hit list returned by a folder query. The possible locations are cache storage, archive storage, and external storage.

Important: Use care when enabling this feature. The display document location function can result in degraded search performance because the storage location information for every document returned for the hit list must be retrieved from the OnDemand object server.

Note Search

The Note Search setting determines when the user will be notified that a note exists for a report document. If the annotation parameter in the application group is set to "No", the Note Search parameter determines when OnDemand searches the database for annotations and notifies the user of the annotations. The possible options are:

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

We recommend that you set the annotation parameter in the application group advanced settings to handle annotation storage and display. When the application group annotation parameter is set to “Yes”, an annotation flag is set in the database when a user adds an annotation to a document. When an annotation exists for a document, a note icon is displayed in the folder document hit list.

Maximum Hits

Maximum Hits (Figure 2-8 on page 38) sets the maximum number of document hit list entries to be returned by a folder query. Limiting the number of hits that can be returned from a query prevents performance degradation that might be experienced if an extremely 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 the OnDemand client can be used to stop the creation of the hit list.

Note: OnDemand does not guarantee the order in which the hits are retrieved from the database. If the hit list size is limited, it is possible that you might not see the most recent documents. If the most recent documents in the application group are required, the query must be qualified in a way that results in a hit list that does not exceed the *maximum hits* parameter. Furthermore, if Load Date is defined as one of the fields in the application group, you can set descending sort for the load date, in that way, documents that are loaded last are listed first.

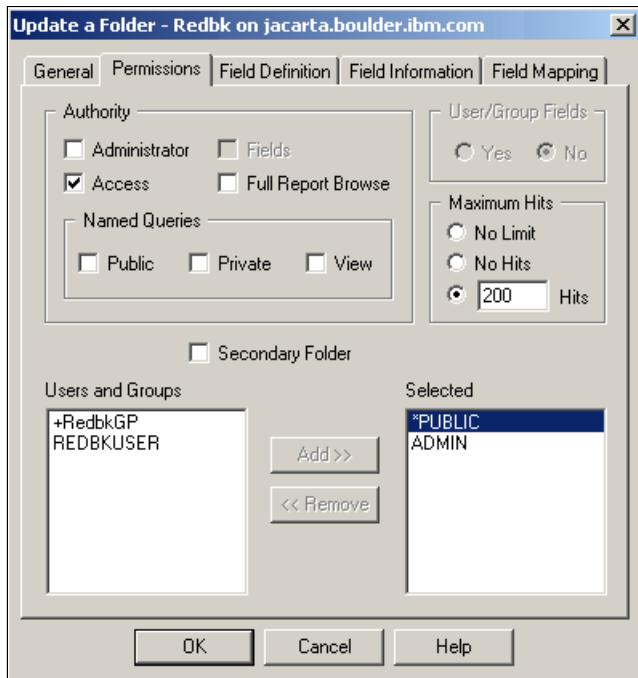


Figure 2-8 Folder permissions

Secondary Folder

The Secondary Folder parameter (Figure 2-8) is used to manage the number of folders that a user is presented with when they log on to the OnDemand system and their list of folders is displayed. By default, 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 option in the OnDemand client.

Text Search

Text Search 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.

Using Text Search allows a user to further qualify a search without adding the overhead 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, 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 do not need to be mapped to database fields.

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. Wild card searches and pattern searches are not allowed. Text search is not case sensitive.

Other text search limitations are based on the type of the documents to be searched and the platform OnDemand is running. For more information, refer to the OnDemand Information Center at the following Web address:

<http://publib.boulder.ibm.com/infocenter/cmod/v8r3m0>

To create a text search field:

1. Create a new folder using the Report Wizard or using an existing folder.
2. Make a copy of the new folder or the existing folder, and rename the copied folder.
3. Add a field, with the Field Type as Text Search, to the copied folder (Figure 2-9).

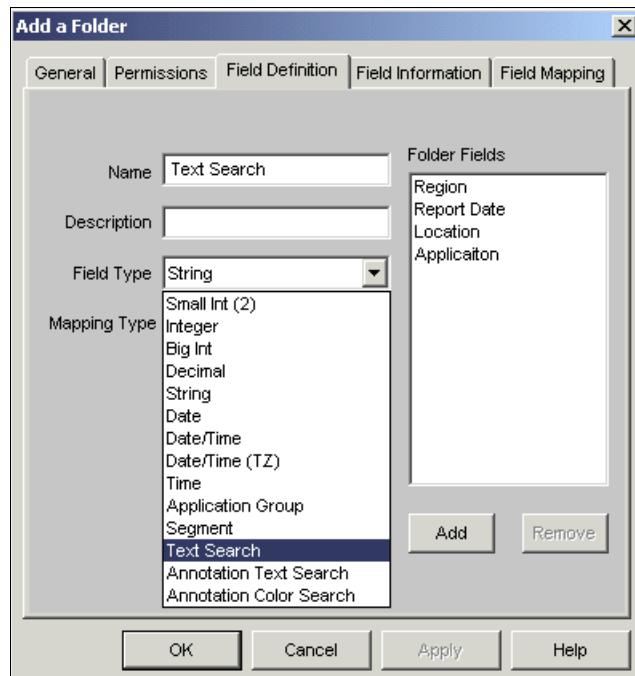


Figure 2-9 Folder field definition: Text Search field

4. Select the **Show Search String** option in the Options menu. This causes the system to highlight the searched text string when the document is opened.

If Autoview in the Options menu is set to First Document or Single Document, the document automatically displays with searched string highlighted. The Single Document option causes the document to be automatically displayed if only one document meets the search criteria. The First Document option causes the first document in the hit list to be displayed automatically with highlighted searched string.

To use the text search field, open the folder with a predefined text search field and perform a text search. When a document returned by a text search is opened for viewing, the viewer is positioned to the first line in the document that contains the text search string. You can use the Find Next option to move to other occurrences of the string in the document.

Note: You can still perform a standard search with this folder. You do not have to specify a text search every time.

One of the advantages of the text search function is that the search is performed on the server. The speed of search is based on the power of the server that is running OnDemand. The disadvantage is the performance hit that might be incurred by the system. The larger the number of documents is that matches the other query fields, the longer it takes for the text search to be performed on this document list in order to build the resulting hit list.

Users should always fully qualify the queries to bring back only the specific documents that they must view. Any sort of wild card search in conjunction with a text search can severely impact performance.

Full Report Browse

Under Folder Permissions, the new Authority option of Full Report Browse has been added to OnDemand Administrative Client (Figure 2-10). This new option allows a user of the Windows client to select a document and retrieve and view the entire report that is loaded with the same load ID.

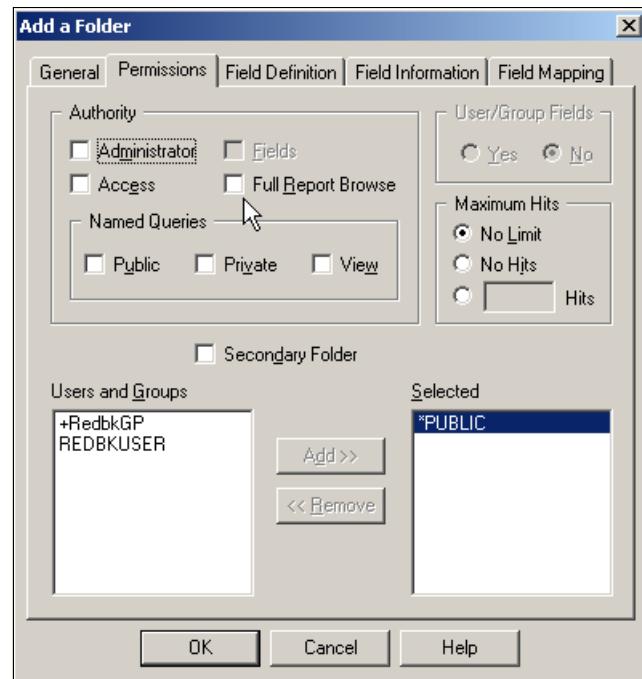


Figure 2-10 Folder Permissions with Full Report Browse for Administrative Client

If the user has Full Report Browse authority for a specific folder, the Windows client has a new View Full Report button, as shown in Figure 2-11. When the user selects the button, OnDemand retrieves the entire report so that the user can view it. If the user does not have the Full Report Browse authority, the button is not visible for that folder in the Windows client.

If the View Full Report button is used, the entire report (with the same load ID) 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.

The screenshot shows the OnDemand Windows client interface. At the top is a 'Search Criteria' dialog box containing fields for Userid, Account, Log Id, Severity, View, Msg Num, Message, and * Time Stamp, each with dropdown menus for operators like Like, Equal To, and Like. To the right of the search criteria are buttons for Search, Clear All Fields, Restore Defaults, Close Folder, and Logical (with AND and OR radio buttons). Below the search criteria is a 'Document List' window displaying a table of log entries. The columns are Lo..., Severity, View, Msg Num, and Message. The data in the table is as follows:

Lo...	Severity	View	Msg Num	Message
29434	Info	No	44	Folder Add: Name(
29434	Info	No	47	ApplGroup to Fold
29434	Info	Yes	44	Folder Add: Name(
29434	Info	No	53	ApplGroup Add: Na
29434	Info	Yes	53	ApplGroup Add: Na
29434	Info	No	39	Group Add: Group(
29434	Info	Yes	39	Group Add: Group(
29434	Info	No	34	User Add: Userid(

To the right of the document list are several buttons: View All Selected, View Full Report (which is highlighted with a mouse cursor), Print All Selected, Sort List..., and Append.

Figure 2-11 Folder view with Full Report Browse authority on a Windows client

2.1.5 The Report Wizard

So far we have discussed how to use OnDemand reporting tools to create an application group, an application, and a folder one by one. There are two ways to define a report to OnDemand:

- ▶ Add a separate application group, an application, and a folder
- ▶ Use the Report Wizard

This section discusses briefly what the Report Wizard can do. For further information, refer to the following Web address:

<http://www.ibm.com/developerworks/db2/library/techarticle/0301wagner/0301wagner.html>

The Report Wizard defines a report to 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 on various windows and by using the graphical indexer to define the indexing parameters, the database fields, and the folder fields.

To start the Report Wizard, you click the **Report Wizard** button located on the main window of the Administrative Client, as shown in Figure 2-12.

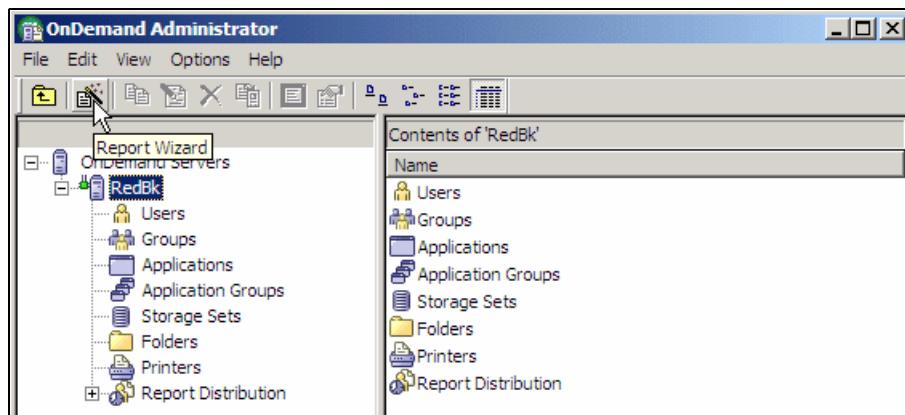


Figure 2-12 Report Wizard button on the OnDemand Administrator Client

After you go through all the report definitions, you are asked if an application identifier is needed (see Figure 2-13).

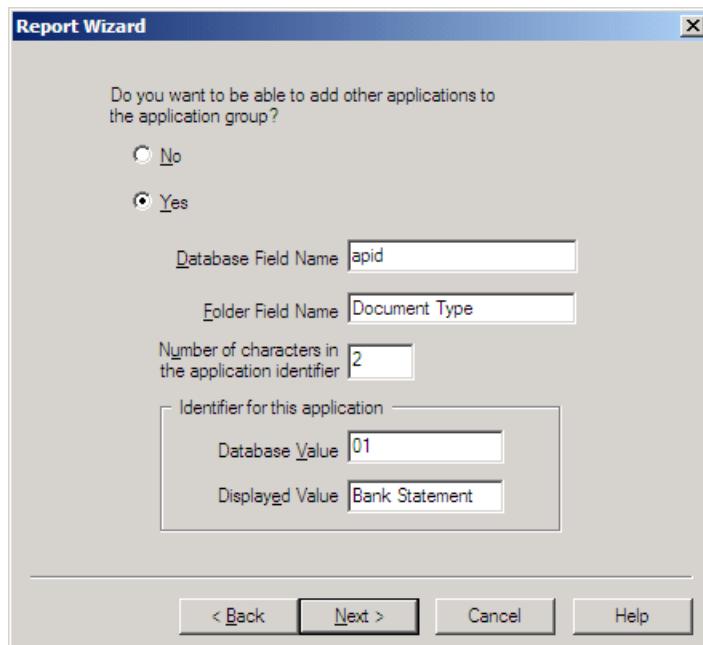


Figure 2-13 An Application identifier Windows with Report Wizard

You are presented with a window on which you name the application group, the applications, and the folder as shown in Figure 2-14. When you complete the Report Wizard, the application group, application, and folder are all created at the same time.

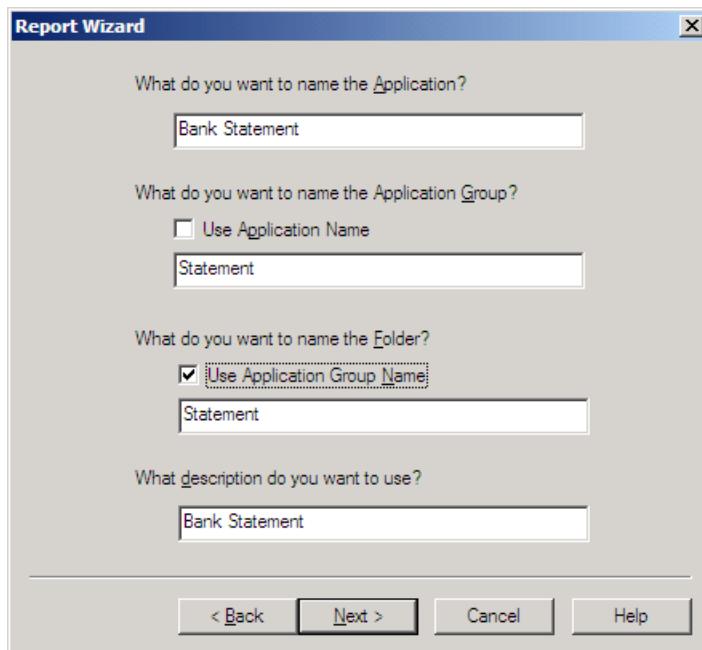


Figure 2-14 Names page of the Report Wizard window

Adding an application to an existing application group

You can also use the Report Wizard to add an application to an existing application group. To add multiple applications to an application group, application identifiers must be available for the new applications. Be sure to add them into the application group first before you start to define the new applications.

To use Report Wizard to add a new application into an existing application group, highlight the application group and click the **Report Wizard** button. If the application group was originally added using the Report Wizard, update the application group so that additional application identifiers can be added for each new application.

Follow the same procedure and answer the questions along. Note that the application group database fields and folder fields are already defined and are not displayed.

The Name field is slightly different as well, because the application group and folder are already defined to the application. From the Application Identifier field, you can choose the Application Identifiers that are defined in the application group previously but are not in use yet.

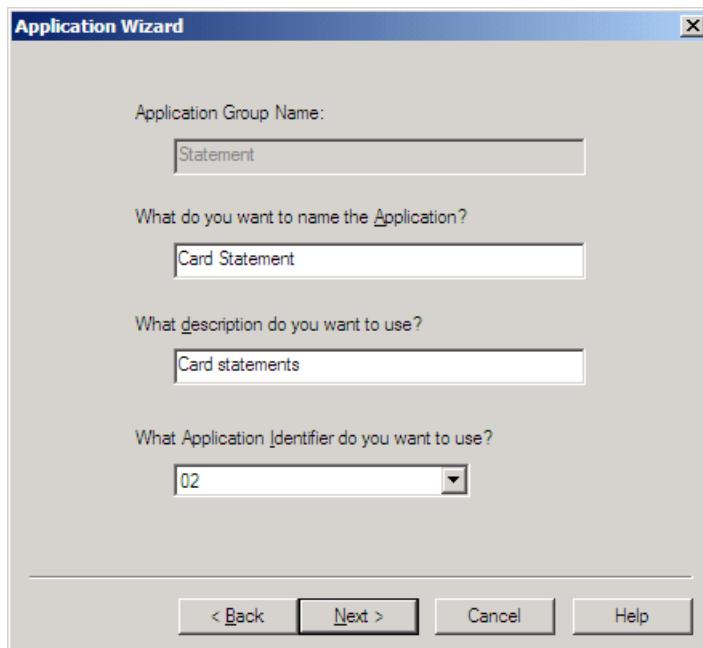


Figure 2-15 Names page of Report Wizard window when adding an application

Note: If AFP is selected as the data type, the report data is line data, which is converted to AFP before it is loaded into OnDemand. The Report Wizard cannot be used to define the report to OnDemand if it is already AFP data.

Summary

Defining a report to OnDemand using the Report Wizard is simple. You answer a few questions and identify the locations of the index values visually in the sample report data. The number of questions that you answer are kept to a minimum and are related to the values that cannot be changed after the report is defined. For any values that are not assigned based on the answers to the questions or by using the graphical indexer, default values are used and changed later by updating the application, application group, or folder.

2.2 User and group administration

When designing an OnDemand system, decisions must be made concerning the best way to implement the many authority structures that are available for users and administrators of the system. The span of control for the administration of the system must be considered along with the level of user access to the data stored in the system. How many different administrators will be 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 carry out their jobs?

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 is typically used when the number of reports and users to be added to the system are 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 will perform all administration tasks.

In a system design with decentralized control, different users are granted different levels of administrative authority. For instance, 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, while 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 OnDemand system but must be maintained independent 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 on 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 chosen can be changed at a later date, the amount of work involved in making that change is greater than the amount of work necessary to study the requirements of the system and put into place the most appropriate administration policies from the beginning.

In this section, we discuss different types of users, followed by a discussion on a decentralized administrative plan. We also introduce a new administrative tool, the OnDemand XML Batch Administration, which is a command line program that is executed on the ONDemand server.

2.2.1 User types, authorities, and functions

There are generally four different types of users in an OnDemand system. Each has a different level of access, authority, and responsibility in the system:

- ▶ **User:** Logs in and query the system to retrieve documents and reports for viewing
- ▶ **User administrator:** Add users or other user administrators to the system
- ▶ **Report administrator:** Define the application groups, applications, and folders to be part of the system

They are responsible for knowing the report and document data and for defining the indexes to be extracted from the data and stored. Report administrators are also responsible for designing the user interface to the reports through the folder definition process and for controlling access authority to the reports that they design, index, and load.

- ▶ **System administrator:** Has the highest level of authority in an OnDemand system

They have 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 grant create group authority, create storage sets, and define system printers.

When the administrative tasks and levels of authorities are understood, decisions must be made concerning the span of control in the system. Is it better to have one user control all access and functions in the 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 the factors that were discussed previously concerning centralized or decentralized administrative control.

As stated earlier, a centralized administrative plan is best suited for an OnDemand system with a small number of users and relatively few reports that must be defined. In the next section, we focus on the decentralized system and discuss the different aspects of a decentralized administrative plan.

2.2.2 Decentralized system administration

OnDemand provides enough flexibility in types of users and in levels of authority to allow many different methods to decentralize administrative control of a system. In most cases, the need to control access to specific data or to control access to specific OnDemand objects is the determining factor in choosing a decentralized administrative model. Next we examine two decentralized administration scenarios. One is the *object type model* and the other is the *object owner model*.

Object type model

The object type model (Figure 2-16) is used to control access to specific objects in the OnDemand system. All users and groups in the system are created and maintained by a user administrator. Reports in the system are created and maintained by a report administrator. The object type model lends itself to the division of administrative tasks based on security requirements or skill level. The administrator that adds users and groups does not necessarily require access to all report data. The skill level required to administer users is much lower than the skill level that is required to analyze data and to create and administer report definitions.

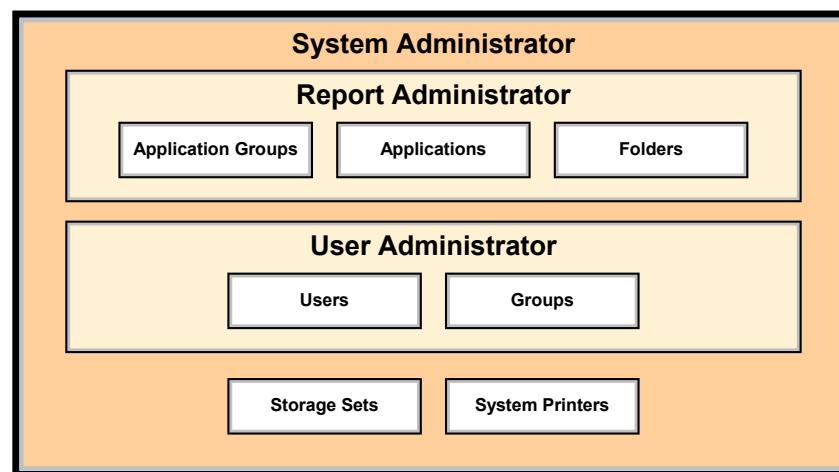


Figure 2-16 Decentralized system administration: object type model

Implementing the object type model

When putting in place an object type model, the OnDemand *system administrator* defines two new users. A *report administrator* is defined and given authority as an application group and folder administrator. A *user administrator* is defined with authority to create users and groups.

The report administrator defines and creates the application groups, applications, and folders that make up the OnDemand system. In this role, the report administrator is responsible for determining the data fields in the report data that make up the indexes for the reports, how the index data is extracted from the report data, how the reports are loaded into the system, and how long the report data and indexes are to be maintained in the system.

The report administrator also controls access to the reports that are created. Access can be granted to individual users or can be granted to user groups. The use of groups for access control simplifies the task. Simply adding a user to a group with access to a specified report prevents the need to grant authority to each user that needs access to a report.

The user administrator adds users to the system and creates groups to which users might be added. Any user added to the group has access to all reports that might be accessed by members of the group. The user administrator also adds the report administrator to the groups that require access to report data. By virtue of being added as a group member, the report administrator can see the group in the permissions list for application groups and folder and can grant access permission to the group. A report administrator does not automatically have access to users and groups when accessing permission lists.

Table 2-1 summarizes different types of the administrators and their corresponding roles.

Table 2-1 Administrator roles in object type model

Administrator type	Administrative tasks
System administrator	Create report administrators Create user administrators with create groups authority Create and maintain storage sets Create and maintain system printers
Report administrator	Create and maintain application groups Create and maintain applications Create and maintain folders
User administrator	Create and maintain users Create and maintain groups

Object owner model

The *object owner model* (Figure 2-17) is a design that allows for report security based on limiting data access to a small group of users. The users and reports on the system are created and maintained by administrators that only work with that group of users and reports. Other users and reports that exist in the OnDemand system are created and maintained by a different set of administrators. This allows the reports and users to exist independently from other groups of users and reports and they are not accessible by other groups in the system.

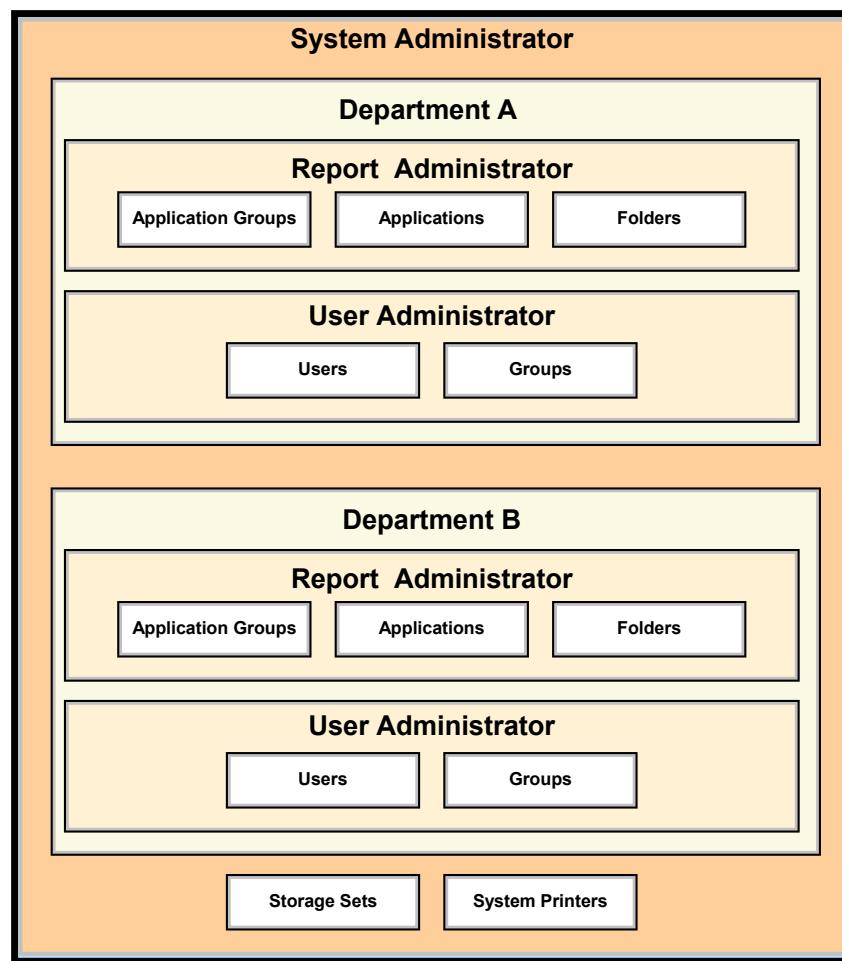


Figure 2-17 Decentralized system administration: object owner model

Departments within the same company are good candidates for the object owner model. Users from one department are completely isolated from the data and

users of another department. There are separate administrators for each department that is being serviced.

Implementing the object owner model

When putting in place an object owner model, use care concerning access to sensitive data. Data from several different sources resides on one OnDemand system. Each group of data must be accessible by a distinct group of users. For each group of data, the OnDemand system administrator defines a report administrator and a user administrator.

The *report administrator* is defined with the authority to create and maintain application groups and folders. The report administrator only has authority over the application groups, applications, and folders that this person creates.

The *user administrator* is defined with the authority to create users and to create user groups. The user administrator only has authority over the users and user groups that this person creates.

The OnDemand system administrator must give the user administrator authority of the report administrator. This authority allows the user administrator to add the report administrator to the groups that have access to the reports that the report administrator is creating. In turn, this allows the report administrator to view and add members of the groups to the permissions list of the application groups and folders that he or she creates.

The responsibilities of the user administrators and report administrators are the same in both the object type model and the object owner model. The difference is that, in the object type model, the administrators have authority over all users and reports in the system, while in the object owner model (Table 2-2), the administrators only have authority over the users and reports that they created.

Table 2-2 Administrator roles in object owner model

Administrator type	Administrative tasks
System administrator	Create a report administrator with create application groups and create folders authority Create a user administrator with create groups authority Create and maintain storage sets Create and maintain system printers
Report administrator	Create and maintain application groups Create and maintain applications Create and maintain folders
User administrator	Create and maintain users Create and maintain groups

Summary

Choosing the right administration model is an important decision in the design of an OnDemand system. Table 2-3 contains general guidelines to take into account when deciding on an administration model.

Table 2-3 Administration guidelines

Environment	Recommendation
The number of reports and users to add to the OnDemand system is small (less 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 OnDemand system. The data must be kept independent of other data in the system. Data must be isolated and access are only allowed for users who must view the data.	Decentralized system administration using the object owner model
Report processing and loading must be limited to a group of users for security reason.	Decentralized system administration using 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 using the object type model

2.2.3 OnDemand XML Batch Administration

In addition to the administrative client that runs under Windows, OnDemand now provides an administrative program that uses Extensible Markup Language (XML). This XML Batch Administration program (XML batch program) is executed on the OnDemand server and provides the same functionality as the administrative client.

The difference between the two programs is that for the administrative client, the user has to provide input through the graphical user interface (GUI) while the XML batch program receives input through the XML interface.

In this section, we discuss:

- ▶ Benefits of using the XML batch program
- ▶ Prerequisite of the XML batch program

- ▶ Using the XML Batch Administrative program
- ▶ Special features of the XML batch program
- ▶ Tips on using the arsxml command
- ▶ Troubleshooting

Benefits of using the XML batch program

There are many benefits of using the XML batch program:

- ▶ It provides another way to perform the 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.
- ▶ Administrative client is not needed.
- ▶ It is useful for replicating the same objects to multiple OnDemand servers, and can even replicate the object when there is no network connection between the servers.
- ▶ It makes automation of system administrative tasks easy.
- ▶ You can analyze or represent the XML output file your own way.
- ▶ For OnDemand support purposes, the output XML file can be used to provide information to the support team for problem determination.

Prerequisite of the XML batch program

The OnDemand Batch System Administration code requires the following products:

- ▶ Java™ Runtime Environment Version 1.4.1 or later
- ▶ Xerces2 Java Parser (originally known as XML4J Parser) Version 2.6.2 or later

The OnDemand Batch System Administration process uses the Xerces2 Java Parser. You must download the parser code before using the Batch System Administration.

The parser performs the following actions:

- ▶ Checks the input XML file for correct syntax
- ▶ Checks the input XML file for valid data objects (based on the schema file)
- ▶ Parses the input XML file and creates internal Java structures that our code can examine
- ▶ Handles different input/output file encoding

You must also create a file named *arsxml.cfg* in the OnDemand *config* directory. This file is used to specify the directory of the Java archive JAR files. The file should only contain one line:

```
ODXMDIR=<dir>
```

In this line, *<dir>* indicates the full path of the directory that contains the Xerces2 Java Parser JAR files, *xercesImpl.jar* and *xml-apis.jar*.

For details about installing the Xerces2 Java Parser, refer to the README.html or README file found in the xml directory of the OnDemand installation root. For AIX, the README files are in the directory /usr/lpp/ars/bin/xml. Follow the instructions in the README file to drop down the parser.

Using the XML Batch Administrative program

This section provides a brief explanation of how to use the new XML batch program. For detailed information, we recommend that you read *IBM Content Manager OnDemand for Multiplatforms - Administration Guide*, SC18-9237. Also read the article “OnDemand XML Batch Administration” on the Web at the following address:

<http://www.ibm.com/developerworks/db2/library/techarticle/dm-0510wagner/>

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

To use the program, you must have the following files, ID, and password:

- ▶ The schema file, *ondemand.xsd*
- ▶ An input XML file (for example, *exportusers.xml*)
- ▶ A user ID and password to access the OnDemand server

In XML, the definition and syntax of the markup language is defined in a *schema file*. For the OnDemand XML batch program, the schema file is called *ondemand.xsd*. It contains the definitions for the OnDemand objects: users, groups, applications, application groups, storage sets, folders, and printers. Each 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*) should 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 OnDemand objects are exported. The XML that is generated can be used as an input for the subsequent **arsxml** command.

Example 2-3 shows a sample of the file exportusers.xml from the xml samples directory. You can change the names of the user name to the users that you want to export.

Example 2-3 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">

    <!-- This will export all of new users -->
    <user name="SAMPLEUSER0" />
    <user name="SAMPLEUSER1" />
    <user name="SAMPLEUSER2" />
    <user name="SAMPLEUSER3" />
    <user name="SAMPLEUSER4" />
</onDemand>
```

You can export objects using the **arsxml export** command. The following command exports the users from the server odserver1 and generates the output file users.xml:

```
arsxml export -u oduser1 -p odpasswd1 -h odserver1 -i exportusers.xml -o
users.xml -v
```

You can import objects using the **arsxml add** command. The following command imports the users who are using the input file users.xml, which can be the generated output file from the previous command, to odserver2:

```
arsxml add -u oduser2 -p odpasswd2 -h odserver2 -i users.xml -v
```

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

```
arsxml delete -u oduser2 -p odpasswd2 -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 using the **arsxml update** command. For example, you want to update the description of the user REDBK with a new description and add the authority to create users. In this case, you construct the XML input file as shown in Example 2-4 on page 57.

Example 2-4 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="REDBK" description="Redbk user" createUsersAuth="Yes" >
        </user>
    </onDemand>
```

The command to update user REDBK is as follows:

```
arsxml update -u oduser2 -p odpasswd2 -h odserver2 -i updateUser.xml -v
```

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

Note: Attributes that have default values are not included in the output XML file. Also when creating an input XML file, not all attributes must be specified for each object.

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 respective 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 do this by specifying the **arsxml** command option **-r d** on the command line.

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

If an error occurs during 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.

Tip: For good performance, the ideal order in the XML file is printers, users, groups, storage sets, application groups, applications, and folders. However, this order is not required. The objects can appear in any order within the XML file.

Tips on using the arsxml command

Before you begin to use the `arsxml` command, we recommend that you read *IBM Content Manager OnDemand for Multiplatforms - Administration Guide*, SC18-9237. Also read the article “OnDemand XML Batch Administration” on the Web at the following address:

<http://www.ibm.com/developerworks/db2/library/techarticle/dm-0510wagner/>

If you are not familiar with the syntax, 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 use to modify to suit your needs. Make sure that the export is successful without any errors; otherwise the output XML file might be incomplete.

Adding objects to the OnDemand server is straight forward. If you are into more advanced operations, such as updating the permission of existing users for an application group or folder, and you are not getting what you are expecting, then you might have missed the `task` attribute. You should include this attribute when you want to update existing object, such as removing a user's permission from an application group or updating a user 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 REDBK from an application group, you must use the following line in the input XML file:

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

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

```
<permission user="REDBK" task="update" queryRes="New Query" />
```

The previous line is incorporated in Example 2-5 for the input file `updateag.xml`.

Example 2-5 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-->
    <applicationGroup
        name="RedbkAG" >
        <permission user="REDBK" task="update" queryRes="Newer Query" />
    </applicationGroup>

</onDemand>
```

The command to update the user REDBK is as follows:

```
arsxml update -hodserver -i updateag.xml -v -u admin -podpasswd
```

Example 2-6 shows the output from the previous command.

Example 2-6 Successful output of updating user REDBK

```
Starting arsxml. Version: 7.1.2.5
Command Line: arsxml update -hodserver -i updateag.xml -v -u admin -p *****
Updating applicationGroup, RedbkAG
Update of applicationGroup, RedbkAG was successful.
Updating applicationGroup-permission, RedbkAG-REDBK
Update of applicationGroup-permission, RedbkAG-REDBK 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 indicating that the object already exists as shown in bold in Example 2-7. In this scenario, user REDBK is not updated with the new query restriction.

Example 2-7 Output of updating the user without using task="update"

```
Starting arsxml. Version: 7.1.2.5
Command Line: arsxml update -hodserver -i updateag.xml -v -u admin -p *****
Updating applicationGroup, RedbkAG
Update of applicationGroup, RedbkAG was successful.
A applicationGroup-permission object named 'RedbkAG-REDBK' already exists.
Finished processing file updateag.xml.
```

Notice that *PUBLIC is part of the basic folder/ application group structure and is automatically added when a folder or application group is created. Therefore, the permissions for *PUBLIC can only be updated; it cannot be deleted or added. You should never specify the task attribute in the <permission> object for *PUBLIC. The task attribute is update by default.

Example 2-8 on page 60 contains a more comprehensive example that shows permission delete, update, and add. It is a portion of an XML file that removes the permissions for users test01-test03, updates the permission values for users test06-test8, and adds the permission for user test11. Use this XML file only during an **arsxml** update operation.

Example 2-8 Delete, update, and add permission sample

```
<applicationGroup name="test07"
    storageSet="test01"
    description=""
    >
    <field name="f1"    appIDField="Yes" />
    <permission user="test01" task="delete" />
    <permission user="test02" task="delete" />
    <permission user="test03" task="delete" />
    <permission user="test06" task="update" docUpdatePerm="Yes" docPrintPerm="Yes"
    docCopyPerm="Yes" annotCopyPerm="Yes"    queryRes="This is a New Query
    Restriction -6"    authority="Logical Views" />
    <permission user="test07" task="update" docUpdatePerm="Yes" docPrintPerm="Yes"
    docCopyPerm="Yes" annotCopyPerm="Yes"    queryRes="This is a New Query
    Restriction -7"    authority="Administrator" />
    <permission user="test08" task="update" docUpdatePerm="Yes"
    docPrintPerm="Yes" docCopyPerm="Yes" annotCopyPerm="Yes"    queryRes="This is a
    New Query Restriction -8"    authority="Logical Views" />
    <permission user="test11" task="add" docUpdatePerm="Yes" docPrintPerm="Yes"
    docCopyPerm="Yes" annotCopyPerm="No"    queryRes="This is a Query Restriction
    -1"    authority="Logical Views" />
</applicationGroup>
```

The README file from the xml installation directory provides a list of technotes that you should review. This README is always updated with the latest technotes that are available. If you have questions or problems with **arsxml**, consult the technotes to see if you can find a solution to the situation that you are experiencing.

Troubleshooting

To test the **arsxml** function while writing this IBM Redbooks publication, we exported the users using the **arsxml** command and received the XML file. We then used the output file, changed the name of the users, and added it back to the server. In doing these actions, we received the following error message:

A parsing error occurred in file oexportdoris.xml, Line 2, Column 111:
cvc-elt.1: Cannot find the declaration of element 'onDemand'.

This message indicates that the schema file ondemand.xsd cannot be found. The *default location* for the schema file is in the current directory. Therefore, when the output file is created, the schema file location is specified as ondemand.xsd. If this is not where the schema file is located, then this line must be changed in the output XML file to specify the full path name before the file can be used as input.

Another issue that we encountered (for OnDemand V7.1.2.5 only) was that, if the export encountered any error, such as a missing user object, the </ondemand> tag is not placed into the output file as indicated in the following example:

```
...
Exporting user, Doris
A user object named 'Doris' does not exist.
Export process completed.
```

In the previous example, we simply added the </ondemand> tag at the end of the output file. Because we were doing testing, we were able to add the tag by ourselves.

This problem is fixed in version 7.1.2.6. In your environment, if you are still using version 7.1.2.5, always check to make sure that the export process runs successfully before you do any import based on the export data.



Database structure

In this chapter, we describe the 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 loading data, show how a search is performed on the database tables, describe the system log, and discuss special considerations for DB2 on OS/390.

In this chapter, we cover the following topics:

- ▶ System control tables
- ▶ Main data table structures
- ▶ Relationship between databases when loading data
- ▶ Search sequence from folder
- ▶ System log
- ▶ Database creation and relationship on z/OS

3.1 System control tables

OnDemand uses system tables and a set of application group data tables. All system control tables are created with the `arsdb` command. The data tables are created when you load data into the OnDemand system.

Table 3-1 shows the OnDemand system control tables with their descriptions.

Note: The complete table name is composed 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 iSeries 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 3-1 OnDemand system control tables

Table name	Purpose	Description
ARSAG	Application group table	One row for each application
ARSAG2FOL	Field mapping table	One row for each application group field mapped to a folder field
ARSAGFLD	Application group field table	One row for each field 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
ARSAGPERMS	Application group permissions table	One row for every user given specific permission to an application group
ARSANN	Annotation table	One row for each annotation added to database
ARSAPP	Application table	One row for each application defined to OnDemand
ARSAPPUSR	User logical views table	One row for each logical view defined for a specific user
ARSFOL	Folder table	One row for every folder defined in OnDemand

Table name	Purpose	Description
ARSFOLFLD	Folder field table	One row for every folder field defined for a folder
ARSFOLFLDUSR	Folder user field table	One row for every field provided for a user given specific field information for a folder
ARSFOLDPERMS	Folder permission table	One row for every user given specific permissions to a folder
ARSGROUP	Group table	One row for each group defined to OnDemand
ARSLOAD	Load table	The load_ID table
ARSNAMEQ	Named query table	One row for each private and public named query defined to OnDemand
ARSNODE	Node table	One row for each storage node defined
ARSPRT	Printer table	One row for each printer defined in OnDemand
ARSPRTOPTS	Printer options table	One row for each printer option
ARSPRTUSR	Printer user table	One row for each user access this 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 defined to OnDemand
ARSUSRGRP	Users in group table	One row for each user assigned to an 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: We recommend that you *do not update* the tables using DB2 system tools such as SPUFI and DB2 Control Center. The tables should only be updated by the OnDemand Administration GUI or OnDemand commands.

3.2 Main data table structures

The OnDemand data tables can grow rapidly. It is important that you understand the structure of the data tables and the relationship between the data tables.

There are two important tables we must examine here, the *segment table* (ARSSEG) and the *application group data table* (ROOT.ag_internal_id). The segment table contains one row for each application group table. Table 3-2 shows the ARSSEG table structure.

Table 3-2 ARSSEG table structure

Column name	Data type	Size	Index	Description
agid	integer	4	Y	Application group ID
table_name	varchar	18	N	Application group segment table name
start_date	bigint	8	Y	Segment start date
stop_date	bigint	8	Y	Segment stop date
post_date	bigint	8	N	Currently not used
closed_date	bigint	8	N	Date table was closed
reimported_date	bigint	8	N	Date table was re-imported
last_update	bigint	8	N	Last update to table
last_backup	bigint	8	N	Last table backup
last_stats	bigint	8	N	Last runstats
mask	integer	4	N	Location
ins_rows	integer	4	N	Inserted rows
upd_rows	integer	4	N	Updated rows
del_rows	integer	4	N	Deleted rows
mod_rows	integer	4	N	Modified rows
max_rows	integer	4	N	Maximum number of rows

Note: The field definitions for ARSSEG might be different depending on platforms. For instance, z/OS does not support the bigint type; therefore, it is an integer.

The start_date and the stop_date are written in an internal OnDemand format. Use the **arsdate** command to get the normal date format. For example, if you get 11992 from the database and use the **arsdate 11992** command, the system returns the date 10/31/02.

The ARSEG 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. It contains a row for each item stored in the application group.

The name of the application group data table is ag_internal_id, which is the identifier that OnDemand assigns to the application group when it is created with the administrative client. The three-digit application group identifier is listed in the Storage Management panel of the administrative client as shown in the example in Figure 3-1. In this case, the application group identifier is EAA.

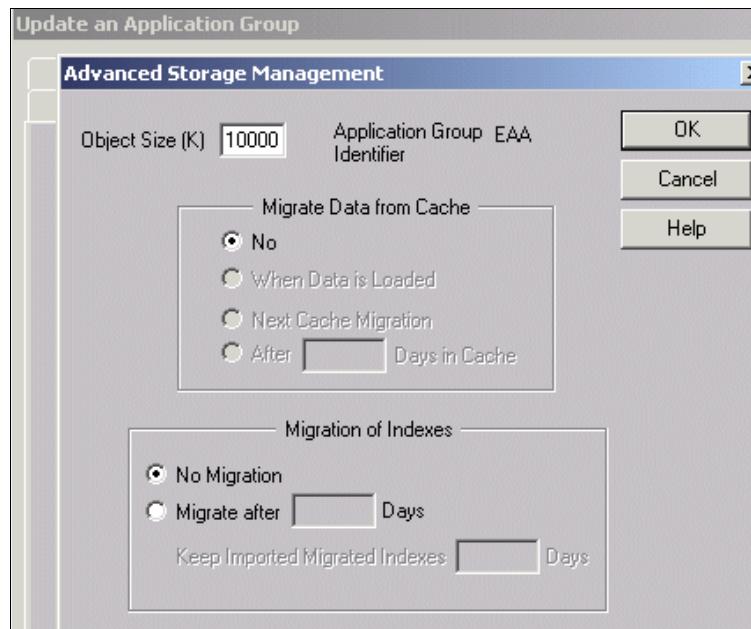


Figure 3-1 Application group identifier example

Table 3-3 shows the application group data table structure.

Table 3-3 AG_internal_id table structure

Column name	Data type	Description
field_1	varies	First user-defined field in application group
field_n	varies	Last user-defined field in application group; you can have up to 32 index field defined in OnDemand
doc-name	varchar	Document name (object name)
doc_off	integer	Document offset in the object
doc_len	integer	Document length
comp_off	integer	Compression offset
comp_len	integer	Compression length
annot	char	Annotation flag
comp_type	char	Compression type
resource	integer	Resource identifier: 0 stands for no resource, and n stands for the resource ID to use
pri_nid	sm-int	Primary storage node identifier
sec_nid	sm-int	Secondary storage node identifier

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

Three more tables might grow rapidly during the lifetime of an OnDemand system:

- ▶ The annotation table (ARSANN), if a lot of annotations are added to the documents
The system creates one row for every annotation. This means every yellow sticker and every graphical annotation add one row to this table.
- ▶ The resource table (ARSRES), if a lot of AFP data is archived in an OnDemand system and the resources such as formdef, page segments, and overlays are often changed
The resource table grows depending on the amount of resources that are added and changed.

- ▶ The load table (ARSLOAD), if a lot of **arsload**, loading data to the system, is done

The system creates one row for each load. The load table (see Table 3-4) can grow to a multimillion row table during the lifetime of an OnDemand system.

Table 3-4 ARSLOAD table structure

Column name	Data type	Description
agid	integer	Application group identifier
pri_nid	smallint	Primary storage node identifier
sec_nid	smallint	Secondary storage node identifier
name	varchar(11)	Name of the load
start	integer	Start date in segment
stop	integer	End date in segment
exp_date	integer	Expiration date

The load ID in the system log or after the **arsload** should look like the following example:

6850-25-0-15FAA-9577-9577

3.3 Relationship between databases when loading data

In this section, we present an example that shows the relationships between the databases when loading data to an OnDemand system. This example is based on a check application that has four index fields defined as *customer_name*, *accout_nbr*, *check_nbr*, and *balance*. There is a one-to-one relationship 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 3-2 on page 70 displays the data created in the ARSAG table; the *agid* is 5018, and the *agid_name* is HAA.

This application is defined to create a new application group data table every 10 million rows. During the data loading, OnDemand uses the *agid* and the *agid_name* to create one row into the segment table (ARSSEG) for every 10 million rows. 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* is composed of the

agid_name from the ARSAG table plus a counter. The *max_rows* in ARSSEG table limits the total number of rows can be stored in the application group data table. The *ins_rows* refers to the actual number of rows stored in the data table.

Figure 3-2 displays the two rows created in the ARSSEG table: one row with table_name HAA1, another HAA2. Both HAA1 and HAA2 are the actual names of the application group data tables that are created. ARSSEG keeps track of the maximum rows and the currently inserted rows; one is 10000000, and another is 326098.

The application group data table contains the *doc_name*, which is the actual container for the individual document. The offset and the document length are also kept in this table. Figure 3-2 shows the first row has an offset of 0, and the second row has an offset of the document length of the first row.

Figure 3-2 shows the relationship between the tables.

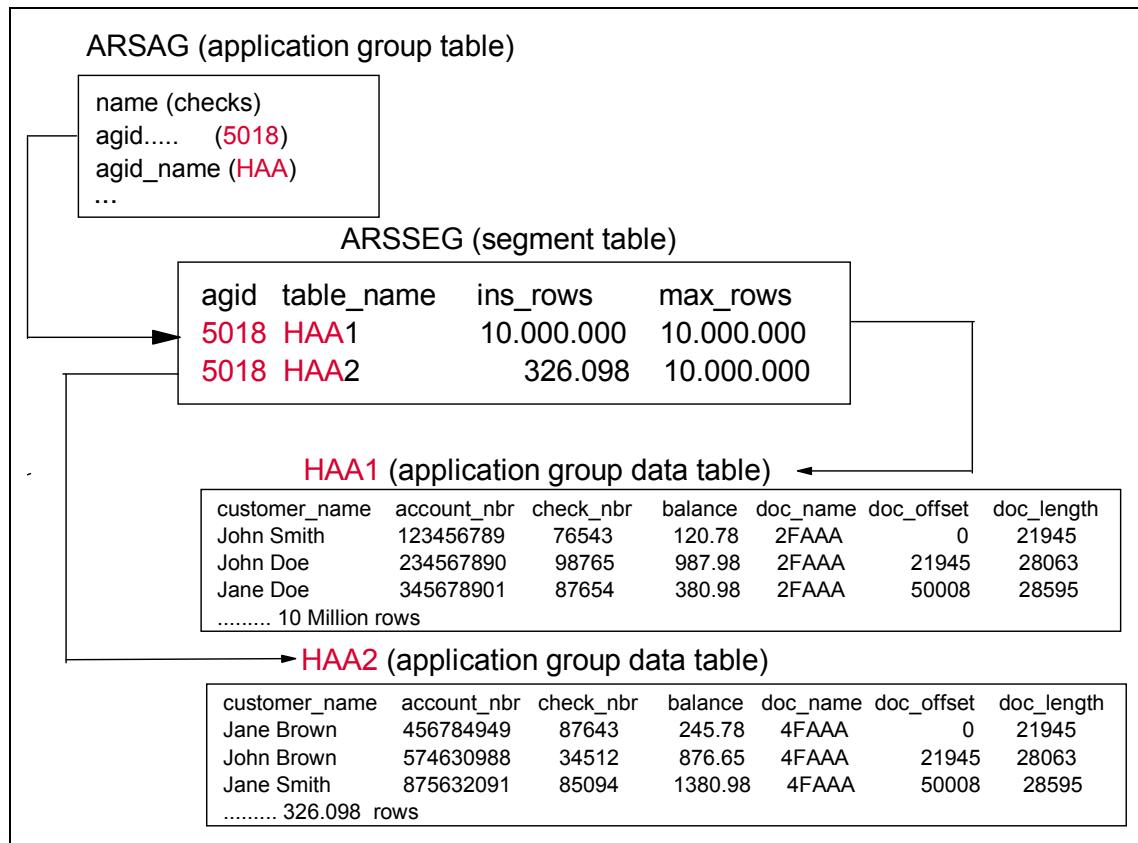


Figure 3-2 Relationship between system tables and data tables

The architecture of relating one application group to one or more application group data table allows OnDemand an unlimited growth of index space. The maximum table size is a limitation of the database subsystem and should be configured for optimal performance. Because this architecture enables a system to create new tables when the maximum table size is reached, there is no logical limitation to the system; rather the limitation is on the physical resources such as processing power, disk space, object servers, and storage hardware.

3.4 Search sequence from folder

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

From the OnDemand standard windows client, a search criteria panel is displayed (see Figure 3-3). In our example, there are four index fields: name, account, Statement Date, and Balance. The example shows a search for a specific date and a balance amount.

The screenshot shows a Windows application window titled "Credit Card Statements - Search Criteria and Document List". The window has a title bar and a main content area. The content area is labeled "Search Criteria" and contains four entries, each with a field name, an operator, and a value. The entries are:

Field	Operator	Value
name	Like	[Empty]
account	Like	[Empty]
Statement Date	Equal To	3.05.1994
Balance	Equal To	104.18

Figure 3-3 OnDemand client search criteria panel

After you enter these values, OnDemand starts searching the folder table, ARSFOL, to determine the folder identification, *fid*. Figure 3-4 on page 72 shows that the *fid* is 5022.

After getting the folder information, OnDemand searches the ARSAG2FOL table for the application groups associated with this folder. Figure 3-4 on page 72 shows the application group ID, *agid*, is 5020.

In general, any number of application groups can be connected with one folder. In our example, there is only one application group associated with this folder.

After getting agid from the ARSAG2FOL table, the system searches through the segment table, ARSEG. Based on the data that is stored in OnDemand internal format in the Database, OnDemand gets the table_name to search for the index values (3.05.1994 and 104,18) in the application group data table (HAA1) and finds the matching *Statement_date* and the *Balance*, and returns these values to the client in a search result list.

To retrieve the individual document within this result list, the system goes back into the application group data table and locates the document offset and data set (object) and retrieves the object for display at the client.

Figure 3-4 shows the details of this search sequence from a folder.

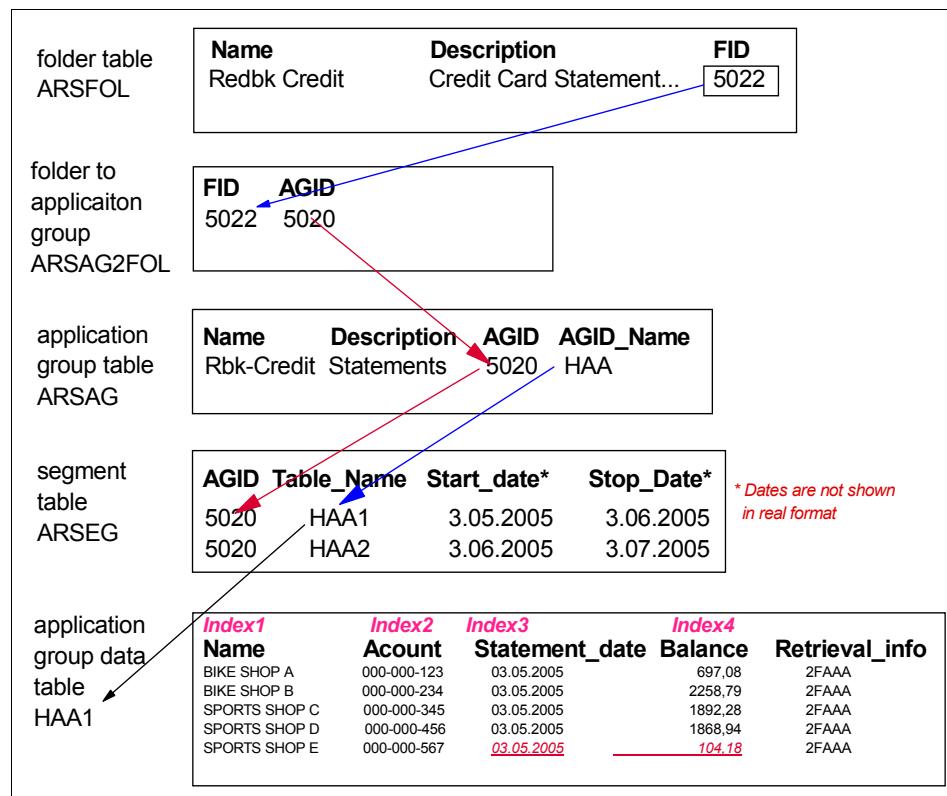


Figure 3-4 Search sequence from a folder

3.5 System log

A *system log* is an application group. It is created with the ARSSYSCR program. The application group identification name is SL and a 4-byte agid is added as well. You will find SLXX in the ARSEG table depending on how big 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 is changeable.

3.6 Database creation and relationship on z/OS

The database creation and the allocation of space for tables and tablespace of the OnDemand product is different in the z/OS environment than the other type of environment. In general, the database administrator (DBA) is responsible for the allocation, creation, maintenance, backup and recovery of the database subsystem. This responsibility is not changed with OnDemand V7 on z/OS.

3.6.1 System tables

Every database transaction is done in a z/OS environment. The standard backup and recovery procedures are used for the OnDemand-created databases and tables. To minimize the effort of creating and monitoring the OnDemand data tables, several automated table creation and space allocation procedures are implemented into the product.

All system tables, as mentioned in 3.1, “System control tables” on page 64, are created with the **arsdb** program. The tables are created in one tablespace. This tablespace is created during the installation with the ARSTSPAC member in the ODADMIN.V7R1M0.SARINST installation file. The size of the tablespace is set there. Before you create the tablespace, you must create the storage group and the database. It is important that the owner of the database (the submitter of the job or the user ID that is set by the “Set current SQLID =’username’”) match the entry SRVR_INSTANCE_OWNER in the ARS.INI file.

Figure 3-5 shows the relationship between the configuration file, the tablespace, and the system tables.

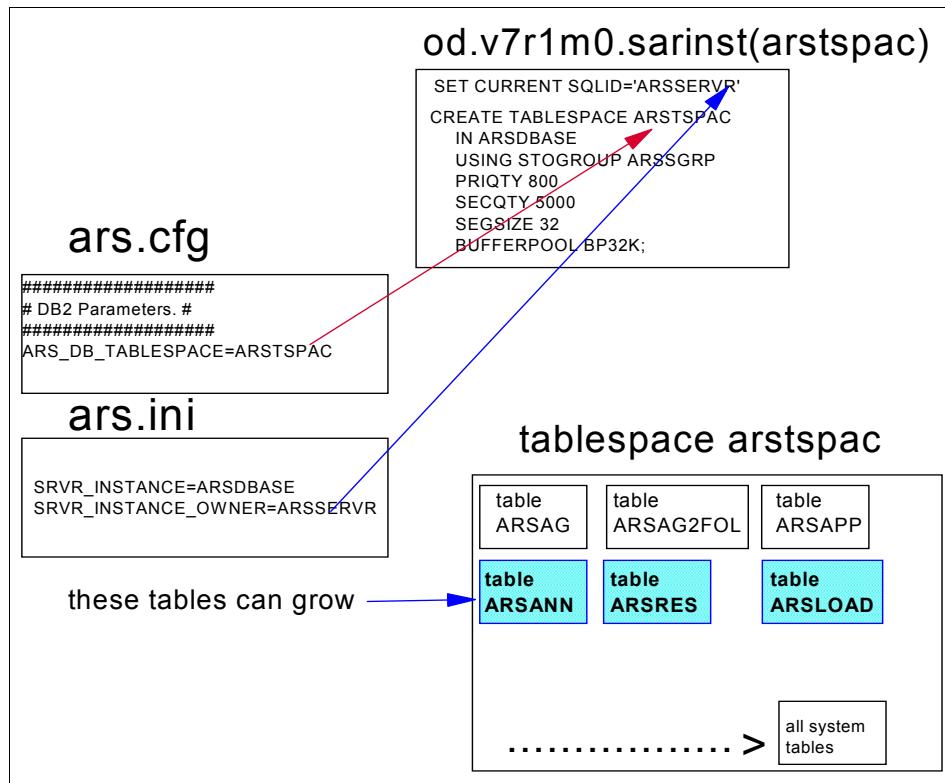


Figure 3-5 Relation between configuration files, tablespace, and system tables

The **arsdb** program provides an interface between the database manager and OnDemand. Several parameters are used in the creation and dropping of tables. For example, to create initial tables, we use the **arsdb -c** command. Refer to *IBM Content Manager OnDemand for z/OS and OS/390 - Configuration Guide*, GC27-1373, for more details.

The **arsdb** program resides in the UNIX System Services file system path `\usr\lpp\ars\bin`. There are several parameters along with the **arsdb** program. Always keep in mind that the **ars.cfg** file is required to determine the tablespace name. Today, there is no way to set the individual size of the system tables because there is only one tablespace associated to them. The creation is done automatically in the background. The only storage allocation size parameter that is changeable is on the create tablespace statement in **arstspac**. The **arsdb** command allows the creation of all or specific tablespaces.

It is possible to create every table individually with the following command:

```
arsdb -c tablename
```

Changing the `ars_db_tablespace`, and create another tablespace with a different name, and assign multiple tablespaces to the system tables. This is possible, but we do not recommend it. The best way to change the tablespace is the `ALTER TABLESPACE` command in DB2.

3.6.2 Data tables

The data tables in OnDemand are created under the control of DB2 on z/OS. Like the system tables, this is done automatically during the `arsload` process. It is important to understand how OnDemand on z/OS is allocating space for these tables, because they can grow very large.

During the creation of an application group, a parameter limits the maximum rows for one table. If this limit is reached, OnDemand creates another data table during the `arsload` process. The maximum row value for an application group table is customized in the Advanced section of the General tab in the AG configuration panels. The field is called *Maximum Rows*.

The allocation of space is done 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 new resources needed for the next couple of months.

Four major factors influence the amount of storage needed for the 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) OnDemand keeps the indexes in the database

Refer to *IBM Content Manager OnDemand for z/OS and OS/390 - Introduction and Planning Guide*, GC27-1438, for information about space requirements.

OnDemand for z/OS and OS/390 allocates its tablespace 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 done in kilobytes. The allocation values depend on the maximum row limit set when creating the application group. The DBSIZE depends on the number of index fields defined in the application.

As a rule of thumb, the calculation of the DBSIZE is as follows:

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, refer to Chapter 8, "Estimating Disk Storage", in *IBM DB2 UDB for z/OS and OS/390 - Administration Guide*, SC26-9931.

Note: Based on this calculation, when you define the application group, make sure that you lower the default of 10 million rows if you only want to store a small amount of data. If you leave the 10-million-row default unchanged, OnDemand allocates 6 million rows at the primary allocation.



Multiple instances

In this chapter, we discuss setting up multiple instances during the installation for different environments. We cover in detail the steps to define and work with a second instance. A separate section is included for each platform: UNIX, Windows NT, iSeries, and z/OS. In addition, we discuss the implications of each configuration.

In this chapter, we cover the following topics:

- ▶ OnDemand instance
- ▶ Multiple instances on UNIX
- ▶ Multiple instances on Windows NT
- ▶ Multiple instances on iSeries
- ▶ Multiple instances on z/OS

4.1 OnDemand instance

An *OnDemand instance* is a logical server environment made up of a database, a library server, and one or more object servers. Traditionally, there is just one OnDemand instance per physical system, but creating multiple instances is a nice way to segment applications.

Creating multiple instances might be a useful way to differentiate between a development environment and a test environment, or to allow a production OnDemand system to run with its own database. Creating a separate instance also allows for having multiple databases, each with their own code page.

4.2 Multiple instances on UNIX

In this section, we describe how to set up multiple instances in a UNIX environment.

4.2.1 Defining a second instance

By default, the initial instance on any library server is archive. By updating the OnDemand configuration files, creating the DB2 instance, defining new tablespace file systems, and defining cache file systems, a second or even a third instance can be defined on the same physical machine.

Here are the steps to create a second instance:

1. “Create a user and home directory” on page 78Create a user and home directory.
2. Create a DB2 instance.
3. Update the configuration files.
4. Create an OnDemand database.
5. Initialize the system log and migration facility.

Create a user and home directory

The first step when defining a new instance is to create the user ID to own the database instance. By default, the initial OnDemand database is *archive* and is owned by the *user archive*. We recommend that you use the same naming convention for each additional instance and use the same name for both the owner and the database instance.

Optionally, you can allow DB2 to automatically create this user when the database instance is created. However you may want to create the user manually, so you can verify that the proper authorities are set before creating the

instance. You may also select to use an existing user. This instance owner must belong to the sysadm1 group. Make a note of the home directory for the user, because this is where the SQL libraries will be installed when the database is created.

The default user and the default instance are archive. In our scenario, we created a user called *ondtest* that owns the ondtest instance. We used **smitty** to create this user and added the user ID to the sysadm1 group.

Create a DB2 instance

There are several methods of creating a DB2 instance; we used the **db2setup** utility. The utility asks the owner and home directory of the database owner. In our scenario, we used the ondtest user ID we created earlier. The group name for the database must be sysadm1.

We recommend that you use the same user name and group name for the fenced UDFs. You see a message warning you not to do this. If OnDemand is the *only* application running on the machine, we recommend that you bypass this warning message.

Update the configuration files

Four configuration files must be updated or created. They are installed with OnDemand at installation time. For the AIX platform, they are located in /usr/lpp/ars/config. For Linux, HP-UX and Sun Solaris, they are located in /opt/ondemand/config. The configuration files are:

- ▶ 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. To update the file, simply copy the *archive section* to a new section and update the newly copied section for the new instance.

Figure 4-1 shows a sample ARS.INI file for our scenario.

```
[@SRV@_ARCHIVE]
HOST=9.17.64.210
PROTOCOL=2
PORT=1450
SRVR_INSTANCE=ARCHIVE
SRVR_INSTANCE_OWNER=root
SRVR_OD_CFG=/usr/lpp/ars/config/ars.cfg
SRVR_DB_CFG=/usr/lpp/ars/config/ars.dbfs
SRVR_SM_CFG=/usr/lpp/ars/config/ars.cache

[@SRV@_ondtest]
HOST=9.17.64.210
PROTOCOL=2
PORT=1441
SRVR_INSTANCE=ondtest
SRVR_INSTANCE_OWNER=root
SRVR_OD_CFG=/usr/lpp/ars/config/ars_ondtest.cfg
SRVR_DB_CFG=/usr/lpp/ars/config/ars_ondtest.dbfs
SRVR_SM_CFG=/usr/lpp/ars/config/ars_ondtest.cache
```

Figure 4-1 ARS.INI file sample

Notice that we created a new section called ondtest, the name of our new instance. Each instance should point to the HOST name or IP address of a physical server. OnDemand differentiates instances by the PORT number. This parameter identifies the TCP/IP port that OnDemand monitors for client requests; each instance must use a unique and unused port. The default port is 0, which points to port 1445. You must choose a different value for each additional instance. We chose an unused port of 1441.

The next parameter that must be changed is the SRVR_INSTANCE parameter. This specifies the name of the database OnDemand used. In our scenario, it is ondtest. Again, we recommend that the instance name and the database name are the same. The SRVR_INSTANCE_OWNER parameter must be the user ID that owns the instance. We recommend that this be root, although it is not mandatory.

Finally, specify the location of the server configuration file, the tablespace file system, and the cache file system for each instance. The parameters are SRVR_OD_CFG, SRVR_DB_CFG, and SRVR_SM_CFG respectively. The cache file system might be shared among instances. If you choose to do so, you can define and use the same file for the SRVR_SM_CFG parameter. In this case, you only have one cache parameter file to update.

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

ARS.CFG

When an instance is started, OnDemand reads the ARS.INI file to determine where the server configuration file is located. Each instance must have its own ARS.CFG file that is determined by the ARS_OD_CFG parameter in ARS.INI. Copy the original ARS.CFG file and modify it appropriately. For our scenario, we created a file named *ars_ondtest.cfg*.

Most of the parameters are same for the multiple instances. The only parameters that must be changed are the database-related ones. You must change your DB2INSTANCE parameter to your new name, and you must change the database path, the primary and the archive log file directories. We recommend that each database resides in its own unique file directory.

The ARS_DB2_DATABASE_PATH variable defines the base file system in which the OnDemand database will reside. The default is /arsdb. In our scenario, we created a file system called */ondtest/arsdb* to hold our database.

The ARS_DB2_PRIMARY_LOGPATH and ARS_DB2_ARCHIVE_LOGPATH parameters define the active and offline archive log files, respectively. These directories should also be unique for each instance.

The group to which the database instance owner belongs must have write access to the database directories specified here. The database instance owner is the user ID that you specified when you created the instance. Verify that the entire database tree (*/ondtest/arsdb** in our scenario) is in the sysadm1 group.

Note: We strongly recommend that each instance uses a different set of Tivoli Storage Manager options files.

Figure 4-2 shows the updated sections of ARS_ONDTEST.CFG for our scenario.

```
#####
# DB2 Parameters (Library Server Only) #
#####
#
DB2INSTANCE=ondtest
#
ARS_DB2_DATABASE_PATH=/ondtest/arsdb
ARS_DB2_PRIMARY_LOGPATH=/ondtest/arsdb_primarylog
ARS_DB2_ARCHIVE_LOGPATH=/ondtest/arsdb_archivelog
ARS_DB2_LOGFILE_SIZE=1000
ARS_DB2_LOG_NUMBER=20
```

Figure 4-2 ARS_ONDTEST.CFG file sample

ARS.CACHE

OnDemand supports cache storage for temporary storage and high-speed retrieval of reports that are stored on the system. Each OnDemand instance can have its own cache storage to allow for a complete differentiation between the instances.

Alternatively, OnDemand instances can share the same cache storage. This is because OnDemand separates the cache directories by first placing the instance name at the cache directory defined. For the archive instance, however, the cache directory is directly below the defined file system name. For the rest of the instances, the cache directories are separated by the instance name. The SRVR_SM_CFG parameter in the ARS.INI file identifies the cache file systems 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 OnDemand stores the control information. After you define this value, you cannot add or remove it from OnDemand or change it in any way.

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 should be similar to the following example:

```
drwx-----    4 root      system      512 Oct 30 12:38 arscache
```

ARS.DBFS

The ARS.DBFS file is called from the ARS.INI file at the instance startup. The ARS.DBFS contains the file names in which OnDemand can store tablespace, and it determines the type of tablespace that OnDemand can create. Storing application group index data in a tablespace is optional, but highly recommended. We also recommend that these file systems contain only OnDemand data and that each instance on the server has its own file systems on which to store data. In general, the more tablespace file systems that you define, the better the system performance is. When using more than one, each of these file systems should have the same allocated disk space.

When using DB2 as the database, OnDemand supports the use of SMS tablespace. Using SMS allows the operating system to increase the size of the tablespace, as required, during a load process.

When creating a new instance that uses tablespace, you must create a new ARS.DBFS file. We created *ars_ondtest.dbfs* in our scenario. Each line in this file must contain the name of the file system and the type of tablespace to be stored. These file systems must be owned by the database instance owner and the group. In our scenario, it is owned by ondtest and belongs to the sysadm1 group. See the following example for the correct permissions:

```
drwxrws---    4 ondtest   sysadm1      512 Dec 27 2001 /arsdb/db1/SMS
```

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

Create an OnDemand database

After the database instance is created, and all the OnDemand directories are set up with the appropriate permissions, it is time to create the OnDemand database. Verify that the group that the database instance owner (ondtest) belongs to has write access to the database directory names 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 OnDemand system tables and indexes
- ▶ Binds the database to OnDemand

Sign on to the user account that you assigned as the owner of the OnDemand instance (in the ARS.INI file). In our scenario, this is root. Run the **arsdb** command with the following options:

```
arsdb -I ondtest -gcv
```

Here -I is the OnDemand instance.

After this command completes, you should be able to log into DB2 and connect to the new instance. List all the tables by typing the following command:

```
db2 list tables for all
```

If you list the tables, you should see the new ARS tables, owned by root. If this command fails for any reason, it creates a db2uexit.err file in the ARS_TMP directory specified in the ARS.INI file; by default, it is /tmp.

Initialize the system log and migration facility

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

```
arssyscr -I ondtest -l
```

Here -I is the new OnDemand instance.

You must also initialize the system migration facility by entering the following command:

```
arssyscr -I ondtest -m
```

Again -I is the new OnDemand instance.

The **arssyscr** program creates the application groups, applications, and folders required by the system logging and system migration facilities.

Note: The **arsdb** and **arssyscr** commands are located in /usr/lpp/ars/bin in AIX and /opt/ondemand/bin in Linux, HP-UX, and Sun Solaris.

4.2.2 Working with the second instance

This section explains how to work with the second instance.

Starting and stopping arssockd

You are now ready to start the new instance. Start the new instance the same way you do the original, but add the instance name after the **arssockd** command:

```
arssockd ondtest
```

Or use the following new command syntax:

```
arssockd start ondtest
```

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

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

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

```
root 33900      1  0 13:02:37      -  0:00 arssockd-onetest: (accepting)
```

The original instance (or archive instance) has no instance name:

```
root 3316      1  0  Feb 27      -  0:00 arssockd: (accepting)
```

Be sure that when you stop the instance, you stop the correct one. You might stop the instance by issuing a **kill** command on the process identifier (PID) of the accepting process or by using the following command:

```
arssockd stop ontest
```

Connecting to instances

To connect to a particular instance, the client must log on to the correct library server. Add a new 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 OnDemand instance to process. You must specify the parameter and the instance name if:

- ▶ 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 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 OnDemand 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 specified is used, for example:

```
arsload -g applicationgroup -u userid -p password -I ontest test.data  
arsmaint -cmcsv -I ontest
```

4.3 Multiple instances on Windows NT

In this section, we describe how to define a second instance on Windows NT environment.

4.3.1 Defining a second instance

Connect to the library server in the OnDemand configurator and select **File → New Instance**. Notice that you must highlight an existing instance in order for this option to show. Select a meaningful name for the new instance.

In the next window, the server panel, click **communications**. Choose a port for the OnDemand clients to use to communicate with the server. You must choose a unique port for each new instance. The default is 0, which defaults to 1445. If you do not change this port, you do not see an error message; instead, every client trying to access the original, archive instance, through port 1445 is now trying to log into this new instance instead. You will be unable to access the original instance until you change this port to a unique number.

We recommend that you define unique file systems for each instance as you define the file systems to control this instance (cache, temp, and database directories). This is a way to keep the instance data and indexes separate from one another. You must assign a unique database directory, primary and archive log file directories, or you will see an error message when creating the database.

After you define the instance, you might click **Create Database Now** to create the DB2 tables. This creates the ARS Db2 tables and initializes the system log and migration facility. You might also choose to run these commands manually by using the **arsdb** command or the **arssyscr** commands with the **-I** parameter.

Note: There are now additional services. There is a new library server, MVS™ download server, and load data server for the new instance.

Refer to 4.2.2, “Working with the second instance” on page 84, for instructions on how to connect to the new instance and how to run the OnDemand programs.

4.4 Multiple instances on iSeries

An OnDemand for iSeries instance is a logical server environment that consists of a server and its own separate database and disk space. An instance is defined in the ARS.INI file.

Each OnDemand for iSeries instance has its own:

- ▶ Set of application groups, applications, folders, and printers
- ▶ Security settings for users, groups, application groups, and folders
- ▶ System log

In addition, each OnDemand for iSeries instance must run in a single Coded Character Set ID.

A second iSeries instance can be created to enable a separate test environment or to physically separate data from two different sources. We recommend that you use the default instance name QUSROND for the first instance.

The QUSROND instance is no longer automatically created with the installation of the Common Server feature. That is because a new parameter was added to specify the locale for the instance. You can specify the language and locale rather than accepting default values that might not be correct for your setup. For instructions on how to create the QUSROND instance, refer to Chapter 12 “Creating an instance” in *IBM Content Manager OnDemand iSeries Common Server - Planning and Installation Guide*, SC27-1158.

The user profile that is used to create an instance must have *SECADM authority and must have the correct locale and locale job attributes set in the profile. Any user profile that is then used to load data into OnDemand also must have the locale parameters set, but does not need *SECADM authority. A profile used to load data should also specify group or supplemental group profiles QONDADM, QRDARS400, and QRDARSADM.

Note: Most OnDemand commands, such as ADDRPTOND and STRMONOND, default to the QUSROND instance name. If other instances are created, the name of the instance must be included in any OnDemand system commands that are instance specific.

When creating a second instance in OnDemand for iSeries, a name must be chosen that is a unique and valid library name for OS/400. No other library by that name can exist. The name *cannot* start with the letter Q and *cannot* be config, CONFIG, www or WWW.

Here are the steps to follow when creating a second instance:

1. Log on to the iSeries server using the CCSID to be used to load data into the instance and issue the following command, changing the parameters for your environment:

```
CALL QRDARS/QRLMINST PARM(REDBK ENU '/QSYS.LIB/EN_US.LOCALE')
```

REDBK is the name of the new instance and is used in the rest of the examples here. *ENU* is the US English language.
'/QSYS.LIB/EN_US.LOCALE' is the locale.

Running this command sets up the second instance:

- Appends the information required for the new instance into the ARS.INI file in the /QIBM/UserData/OnDemand/CONFIG directory
- Creates the new instance directory under /QIBM/UserData/OnDemand
- Creates the ARS.CFG, ARS.CACHE, and ARS.DBFS files within the new instance directory
- Creates the library and database tables for the new instance
- Creates the directories needed by the new instance as specified in the ARS.CFG and the ARS.CACHE files
- Creates a user profile with the name of the instance (REDBK)
- Creates an authorization list with the name of the instance (REDBK)

As the command runs, you see messages for the database table creations and other functions that are being performed. There is a completion message stating that the new OnDemand instance has been created.

2. Edit the ARS.INI file (Figure 4-3 on page 89) using the following command:

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

Page down until you find the section that was inserted into the ARS.INI file for the new instance that was created. The port number must be changed to a port that is not already in use by another instance. By default, a port number of 0 is used for new instances. This causes OnDemand to attempt to use the default port number of 1445, which might already be in use by the QUSROND default instance.

To identify an unused port number, run the following command and press F14 to see which port numbers are listed under LOCAL PORT:

```
WRKTCPSTS *CNN
```

Note: If you are also using the Spool File Archive feature of OnDemand, you must change the port number for the QUSROND instance to something other than 0, for example, 1450. That is because Spool File Archive also uses port 1445.

3. If you do not want to use OS/400 security for the new instance, change the security setting to 0 by modifying the following line in the ARS.INI file (Figure 4-3).

```
SRVR_FLAGS_SECURITY_EXIT=0
```

The default setting of 1 causes OS/400 security to be used for the instance. By choosing not to use OS/400 security, every OnDemand user does not need an OS/400 ID. The user only needs an OnDemand user ID.

```
Edit File: /QIBM/UserData/OnDemand/CONFIG/ARS.INI
Record : __501    of __687 by _10
Control : _____  
  
CMD ....+....1....+....2....+....3....+....4....+....5....  
_____[@SRV@_REDBK]  
____HOST=LOCALHOST  
____PROTOCOL=2  
____PORT=1470  
____SRVR_INSTANCE=REDBK  
____SRVR_INSTANCE_OWNER=QRDARS400  
____SRVR_FLAGS_SECURITY_EXIT=1  
____SRVR_OD_CFG=/QIBM/USERDATA/ONDEMAND/REDBK/ARS.CFG  
____SRVR_DB_CFG=/QIBM/USERDATA/ONDEMAND/REDBK/ARS.DBFS  
____SRVR_SM_CFG=/QIBM/USERDATA/ONDEMAND/REDBK/ARS.CACHE
```

Figure 4-3 ARS.INI file sample for the iSeries server

4. Edit the ARS.CFG file (Figure 4-4) using the following command:

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

If you want the new instance to start automatically when the STRTCPSVR command is issued, change the ARS_AUTOSTART_INSTANCE parameter from 0 to 1.

Do not modify any other values in the ARS.CFG and ARS.INI files.

Note: You can also edit the ARS.INI and ARS.CFG files by mapping a drive to the iSeries integrated file system root directory using Windows Explorer. Open the file in an editor such as Notepad or WordPad, make the desired changes, and save the file.

```
Edit File: /QIBM/UserData/OnDemand/REDBK/ARS.CFG
Record : ____46   of ____62 by _10
Control : _____
CMD ....+....1....+....2....+....3....+....4....+....
____ #
____ ARS_NUM_DBSRVR=5
____ #
____ # AUTOSTART SERVER WHEN USING STRTCPSVR COMMAND
____ # - SET TO 1 TO AUTOSTART THIS INSTANCE
____ #     SET TO 0 TO NOT AUTOSTART THIS INSTANCE
____ #
____ ARS_AUTOSTART_INSTANCE=1
____ #
```

Figure 4-4 ARS.CFG file sample for the iSeries server

5. Start the OnDemand servers using the following command:

```
STRTCPSVR *ONDMD
```

If you want to start only the server for the new instance, issue the following command:

```
Call QRDARS/QRLMCTL *STRTCPSVRREDBK
```

You can confirm that the REDBK instance is started by running the command:

```
WRKACTJOB JOB(REDBK)
```

The job REDBK runs the program ARSSOCKD, which is the OnDemand library server program.

OnDemand jobs use the QRDARS/QOND400 job description, which submits jobs to the job queue QSYS/QSYSNOMAX. That job queue is in the

QSYSWRK subsystem. If the job description has not been changed, you can find OnDemand jobs by entering the following command:

```
WRKACTJOB SBS(QSYSWRK)
```

6. Verify that the instance is functioning correctly by accessing the instance with the OnDemand client. Configure a connection to the instance in the OnDemand client (Figure 4-5) by using the host name or the IP address of the iSeries along with the port number that you assigned to the instance.

User ID QONDADM is automatically added to the new instance. The default password for this user is qondadm1. If the password has been changed for use with another instance, you must use that password. Log on to the server using this ID and the password.

The system log and system migration application groups, applications, and folders are created when the instance is created. Select the system log and run a query for the current day's activity. If you can successfully view the log entries, the instance is set up and communicating correctly.

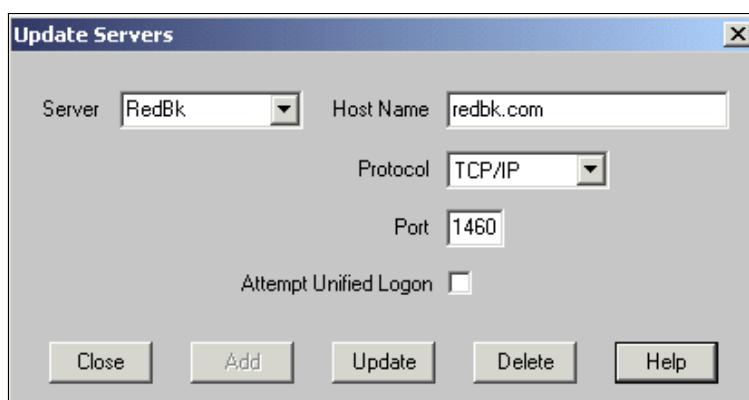


Figure 4-5 Client server setup for the iSeries server

4.5 Multiple instances on z/OS

Instances on z/OS do not differ greatly from those on multiplatforms. The concept is the same. In this section, we explain how to set up a new instance and provide some background information about the UNIX System Services implementation.

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.

An OnDemand instance on a z/OS server is a separately started task (ARSSOCKX) using different databases, users, and application groups. Every user on the instance must be defined for the instance. Every instance has its own security as long as internal security is used. If an external security exit is used, it is common over all the instances. Figure 4-6 shows an overview of the instances on z/OS.

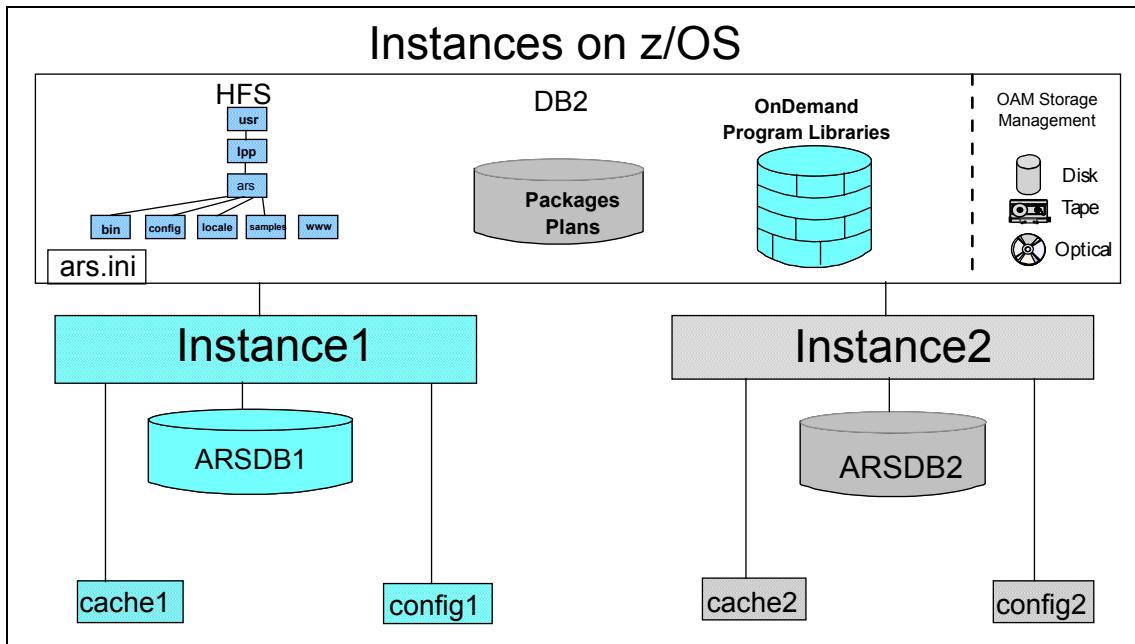


Figure 4-6 Multiple instances overview on z/OS

Each additional instance requires additional system resources such as main storage, virtual storage, and disk space. The administration effort increases with every additional instance. The ARS.INI must be consistent and maintained correctly for all instances. Before you update the ARS.INI, make a copy of the file.

In UNIX System Services, there are many ways to create a copy of a file. Sometimes there are problems with authorization. If you are a superuser, which you can verify by typing `su` on the Open OMVS shell, you can call your systems programmer and RACF® administrator to get the right permissions. When you are in the OMVS shell, you can make a backup copy of any file by using the `copy` command. The following copy options are available:

- ▶ Copy one file to another file in the working directory

- ▶ Copy one file to a new file in the working directory
- ▶ Copy a set of directories and files to another place in your file system
- ▶ Copy a UNIX file to an MVS data set
- ▶ Copy an MVS data set to a file system
- ▶ Copy an MVS data set to an MVS data set

Note: Usually it is sufficient to simply use the following command from the OMVS command line:

```
cp ars.ini /u/ussdflt/arsini.back
```

Here, /u/ussdflt/ is the directory for the copied dataset. It can be any directory with write permissions. For more information about any UNIX System Services command, refer to *UNIX System Services Command Reference*, SC28-1892.

After the dataset is copied, the ARS.INI file can be updated.

4.5.1 Understanding file systems in UNIX System Services

Before we continue with the creation of instances on a z/OS system, we first introduce the UNIX System Services file system on a z/OS system. For an MVS system, the file hierarchy in UNIX System Services is a collection of hierarchical file system (HFS) datasets. Each HFS dataset is one file system and is called a *mountable file system* because you can mount and demount it. A file system is created with the ISPF 3.2 allocate function or with a batch job. Figure 4-7 lists the JCL that is used to create an HFS file.

```
//ALLOCHFS JOB (????,????), 'JOHN SMITH', MSGCLASS=0, CLASS=A,
// NOTIFY=TEAM5, REGION=64M
//***** ALLOCATE HFS FILE *****
//ALLOC EXEC PGM=IEFBR14
//HFSCA1 DD DSN=TEAM5.V710.DBSRES.SERVER.CACHE1.HFS,
// DISP=(NEW, CATLG, DELETE),
// DSNTYPE=HFS, DCB=(DSORG=P0),
// SPACE=(CYL,(100,1,1)),
// UNIT=SYSDA
//*
```

Figure 4-7 JCL used to allocate an HFS file in z/OS

Another important point on UNIX System Services concerns the UNIX permission bits for files in an HFS. Any information about a file is stored with the file in the file system. In a z/OS environment, this information, such as file size

and creator, are stored in a user catalog or VTOC. In a UNIX system, all files have three types of permissions:

- ▶ Read (displayed as r)
- ▶ Write (displayed as w)
- ▶ Execute (displayed as x)

Every permission for a UNIX file (read, write, and execute (rwx)) is maintained for three different types of file users:

- ▶ The file owner
- ▶ The group that owns the file
- ▶ All other users

UNIX files

To determine the permissions for a file, use the `ls -l` command from the command line of the OMVS shell. The following information is returned:

```
-rwxrwxrwx 1 SYSADM1 USERID      203 Jun 28 14:02 ars.ini
```

In this case, the list file and directory attributes command is used for the ARS.INI file (similar to a `dir filename` command in Windows). The `-l` parameter gives you more detailed information about the file.

In this example, the result has the following meaning:

- ▶ `-nrwxrwxrwx`: Permission bits
- ▶ `1`: Number of links to the dataset
- ▶ `SYSADM1`: Name of the file owner
- ▶ `USERID`: Group that owns the file
- ▶ `203`: Size of the file in bytes
- ▶ `Jun 28 14:02`: Date and time the file was last changed
- ▶ `ars.ini`: Name of the file

Note: This example is taken from the IBM Redbooks publication *OS/390 Version 2 Release 6 UNIX System Services Implementation and Customization*, SG24-5178, which is a good reference if you are starting with UNIX System Services.

Permission bit structure

The structure of the ten-character Permission byte field is:

`tffffggooo`

Table 4-1 on page 95 explains the meaning of permission byte `t`. In this structure, `fff` stands for OWNER permissions, `ggg` stands for the GROUP permissions, and `ooo` stands for the OTHER permissions.

Table 4-1 Permission byte information

f	Type of file or directory
-	File
c	Character special file
d	Directory
l	Symbolic link
p	FIFO special file
e	OS/390 LOAD Module

HFS file data is byte-oriented, which differs from the MVS record-oriented datasets. The I/O is done by the use of data stream and not by writing records to it. A UNIX System Services file system on z/OS looks similar to a Windows file system, except for the direction of the slashes. OnDemand expects certain files to be in a specific directory. In UNIX System Services, the root file system is the first file system that is mounted. You do not add application data to this file system.

The path for the OnDemand system is /usr/lpp/ars/. From the ars directory, there are several directories that contain the OnDemand files and executable files, such as programs and procedures. The directories are created at the installation time when running the ARSMKDIR REXX™ routine from the install library, ODADMIN.V7R1M0.SARINST. The /usr/lpp/ars/ directory contains the subdirectories listed in Table 4-2.

Table 4-2 Subdirectories of /usr/lpp/ars

Directory	Contains
bin	All executables, such as arsdb for creating the database
config	All configuration datasets, such as ARS.INI
locale	All subdirectories for national language support (NLS)
samples	All sample files for updating
www	All subdirectories for OnDemand Web Enablement Kit (ODWEK)

Figure 4-8 shows the OnDemand HFS file structure in UNIX System Services.

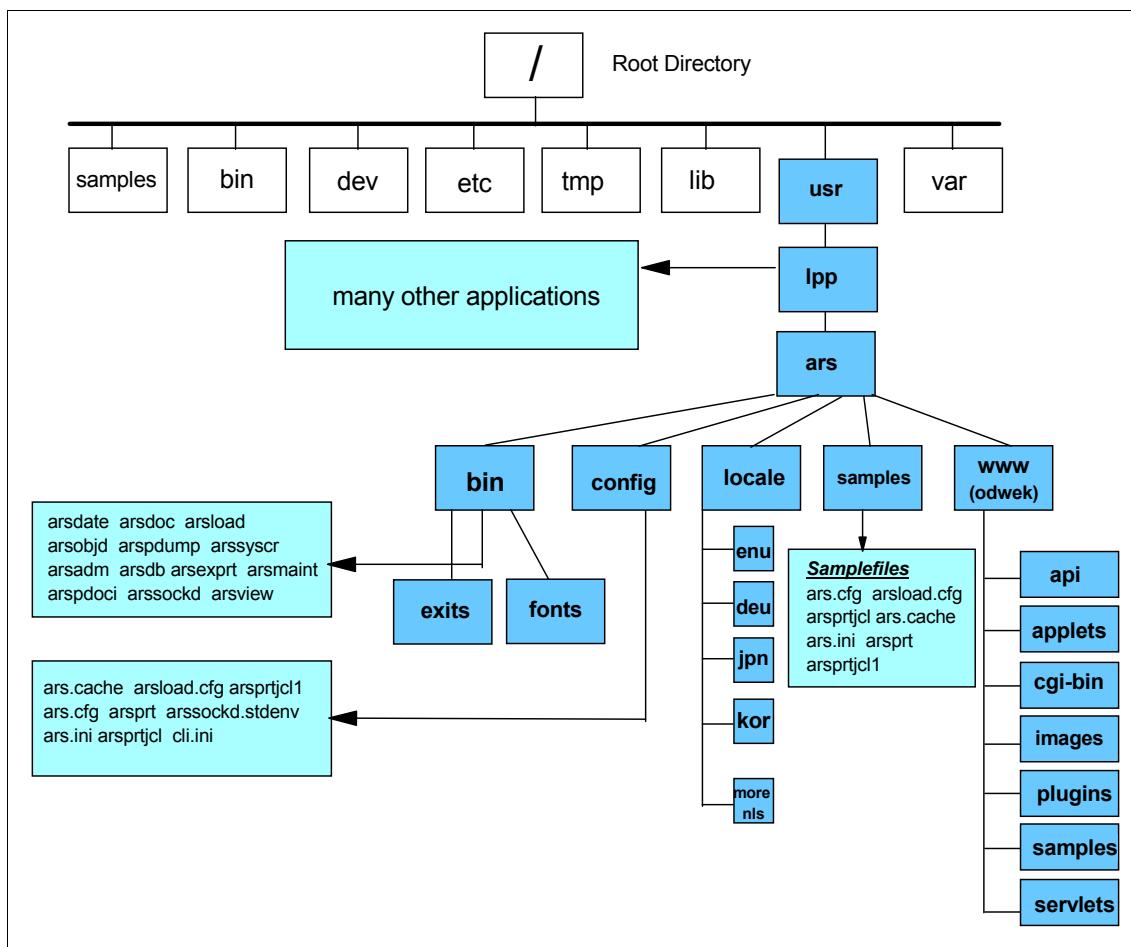


Figure 4-8 OnDemand file structure in UNIX System Services

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

Sometimes when choosing a directory such as /usr/lpp/ars/bin, you see a different path when you issue the **pwd** command. This is 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 HFS. Typically, it contains the name of an MVS data set.

4.5.2 Creating an instance on z/OS

Now that you have a better understanding of the UNIX System Services file structure, in this section we explain how to create an instance on z/OS system.

Adding a file system for the new instance

After a file system is allocated, it must be “connected” with the **mount** command. The **mount** command can be issued in the shell or via a TSO command.

Figure 4-9 shows this command and its relationship to other files.

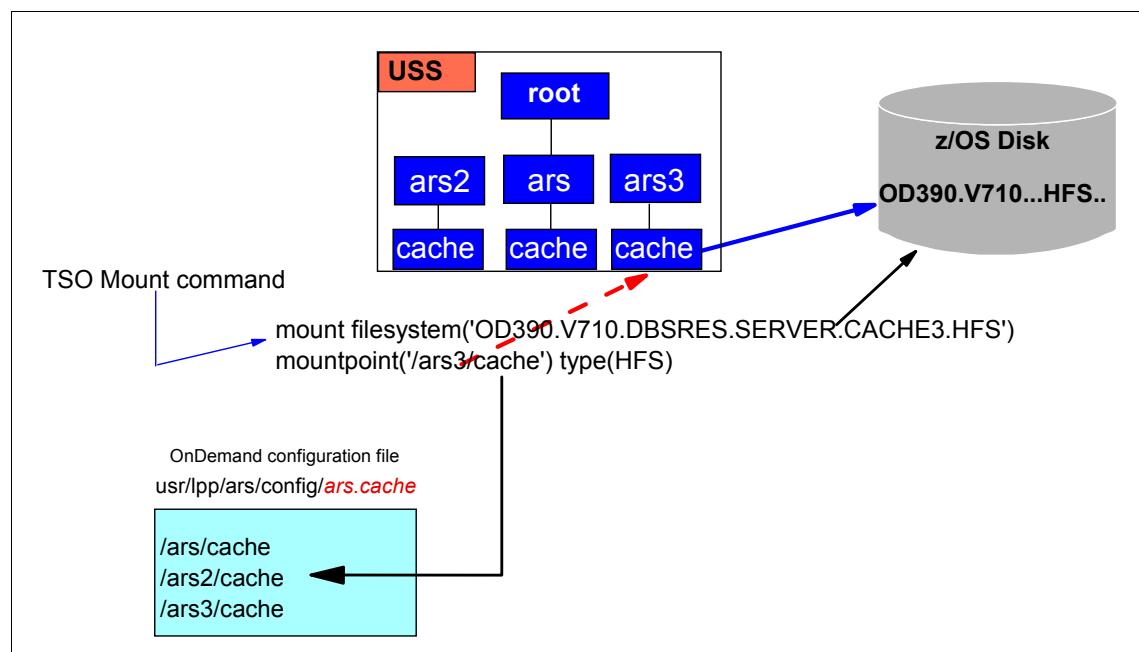


Figure 4-9 Mount of a file system

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. This can be done by modifying the ARSDB2 member in the ODAMIN.V7R1M0.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.
- ▶ Change the Create Storage Group Statement if you want a new storage group for the instance (this is optional).
- ▶ Change the Create Database Statement.
- ▶ Run the job.

Creating the tablespace

The new instance uses its own tables. A tablespace is needed. See 3.1, “System control tables” on page 64, for a detailed description. This can be done by the modification of the ARSTSPAC member in ODAMIN.V7R1M0.SARSINST library, which is used for the initial installation. Modifications to the job are as follows:

- ▶ Change the SQLID to the same user used for creation of 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 a new configuration file

When an instance is started, OnDemand reads the ARS.INI file to determine where the server configuration file is located. Each instance must have its own configuration file, such as ARS.CFG, that is determined by the ARS_OD_CFG parameter in ARS.INI.

Copy the original ARS.CFG file and modify it appropriately. Get the right permission byte sets. In our scenario, a new configuration file, arsins1.cfg (Figure 4-10 on page 99) is created. The important parameters that are database related must be changed:

- ▶ **ARS_DB_TABLESPACE:** The name of the tablespace created with the ARSTSPAC member of the installation library for the new instance
- ▶ **DB2INSTANCE:** The name of the database created with the ARSDB2 member of the installation library for the new instance

All other parameters remain the same, unless you want to try something else with this new instance, such as using object access method (OAM) or a different language.

```

# OnDemand Parameters. #
ARS_NUM_LICENSE=10
#
ARS_LANGUAGE=ENU

ARS_SRVR=
ARS_LOCAL_SRVR=
ARS_NUM_DBSRVR=4
ARS_TMP=/tmp
ARS_PRINT_PATH=/tmp
DB_ENGINE=DB2
#####
# DB2 Parameters. #
#####
ARS_DB_TABLESPACE=ARSTSIN1
DB2INSTANCE=ARSDBAS1

ARS_NUM_OAMSRVR=1
ARS_OAM_DB2SSID=IDB2
ARS_OAM_PLAN=CBRIDBS

```

Figure 4-10 The arsins1.cfg file for the new instance on z/OS

Creating a new cache configuration file

Every instance should use its own file system. Copy the ARS.CACHE file and modify it to be the new instance cache configuration file. Add the formerly created and mounted HFS (in this case /ars/cache3).

Adding the new instance to the ARS.INI file

OnDemand looks up into the ARS.INI file (Figure 4-11 on page 100) to find its parameter values. There is one entry for every instance in the ARS.INI file. These are the parameters that must be changed:

- ▶ **@SRV@_ARSSOCKET**: The instance name
- ▶ **SRVR_INSTANCE**: The name of the database created with the ARSDB2 member of the installation library
- ▶ **SRVR_INSTANCE_OWNER**: The name of the SQLID when the database is created with the ARSDB2 member of the installation library
- ▶ **Port number**: A unique number that is used by instances
- ▶ **SRV_OD_CFG**: The name of the previously created configuration file
- ▶ **SRV_SM_CFG**: The name of the cache configuration file

```
[SSRVS_ARSSOCKD]
HOST=wscmvs.washington.ibm.com
PROTOCOL=2
PORT=1444
SRVR_INSTANCE=ARSDBASE
SRVR_INSTANCE_OWNER=ARSSERVER
SRVR_OD_CFG=/usr/lpp/ars/config/ars.cfg
SRVR_SM_CFG=/usr/lpp/ars/config/ars.cache
[SSRVS_ARSSOCKT]
HOST=wscmvs.washington.ibm.com
PROTOCOL=2
PORT=1555
SRVR_INSTANCE=ARSDBAS1
SRVR_INSTANCE_OWNER=ARSSERV1
SRVR_OD_CFG=/usr/lpp/ars/config/arsins1.cfg
SRVR_SM_CFG=/usr/lpp/ars/config/arsins1.cache
```

Figure 4-11 The ARS.INI file for two instances

Attention: The ARS.INI file is sensitive to the kind of square brackets that you use as a delimiter. Even if it looks the same on the **Ishell** editor when displaying them, it depends on the code page used by the machine, and the Hex value might not represent the correct value, which can lead to unpredictable results.

Example 4-1 shows the correct Hex values for new instance name.

Example 4-1 Hex values for instance name

```
Ý$SRV$_ARSSOCKD"
A7EDE76CDEEDCDCB
DC295CD19226324D
```

Be sure that your bracket is X'BD'.

Creating tables for the new instance

After all the DB2 objects are created and the configuration files are updated, the database for the instance (the system tables) must be created. This is done with the **arsdb** program.

Important: When you work with more than one instance, you must identify the instance name when running the OnDemand programs, such as **arsdb**, **arsload**, and **arssockd**, and for executing database commands.

Create the tables by following these steps:

1. Go into OMVS.
2. Switch to superuser (SU).
3. Set the environment variable to access the DB2 on z/OS.

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

The minimum parameters given are the DB2 SSID and the interface (DSNCLI).

4. Issue the SET command from the OMVS command line.
5. Move to the OnDemand executable directory.

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

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

```
arsdb -I ARSSOCKT -c
```

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

You might see a message similar to the following example:

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

This might look like a DB2 error. Actually, the ARSDB program cannot read the configuration file. Check the log for any RACF messages writing to or opening the file system.

Many installations run several DB2 systems on the z/OS logical partition (LPAR). Sometimes, this 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 with the **export** command:

```
export STEPLIB=ICCDDB2.SDSNEXT:ICCDDB2.SDSNLOAD This 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 with the **SET** command.

Initializing the system log

After you create the OnDemand system tables, the system log must be initialized with the ARSSYSCR program for this new instance:

1. Move to the OnDemand executable directory:

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

2. Run the ARSSYSCR program for this instance by using the -I parameter:

```
arssyscr - I ARSSOCKT -1
```

Here, *ARSSOCKT* is the name of the instance.

Starting the new instance

When everything is set up, you can start the new instance by running another started task on the z/OS system. The JCL is the same, except that it tells the started task which instance to take, by the Parm parameter in the EXEC statement:

```
Parm=('/IIIIIIII XXXXXXXX')
```

In this statement, note the following explanation:

- *IIIIIIII* is the instance name.
- *XXXXXXX* is the program name.

Figure 4-12 shows an example of starting a second instance.

```
//ARSSOCKT JOB (QFTA0000,B123),  
//      'JOHN SMITH',MSGCLASS=0,CLASS=U,  
//      NOTIFY=&SYUID,USER=TEAM5,TIME=1440  
//*****  
//ARSSOCKT PROC  
//ARSSOCKT EXEC PGM=ARSSOCKD,PARM=( '/ARSSOCKT ARSSOCKD' ),  
//      REGION=0M,TIME=NOLIMIT  
//  
//STEPLIB  DD DISP=SHR,DSN=OD390.V710.DBS.SARSLOAD  
//          DD DISP=SHR,DSN=ICCDB2.SDSNEXIT  
//          DD DISP=SHR,DSN=ICCDB2.SDSNLOAD  
//DSNAOINI DD PATH='/etc/ars/cli.ini'  
//SYSPRINT DD SYSOUT=*  
//SYSOUT   DD SYSOUT=*
```

Figure 4-12 Starting a second instance

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

Figure 4-13 provides an overview and the relationships between the steps for the creation of a second instance. You can have as many instances as you want, depending on the resources that your z/OS system has.

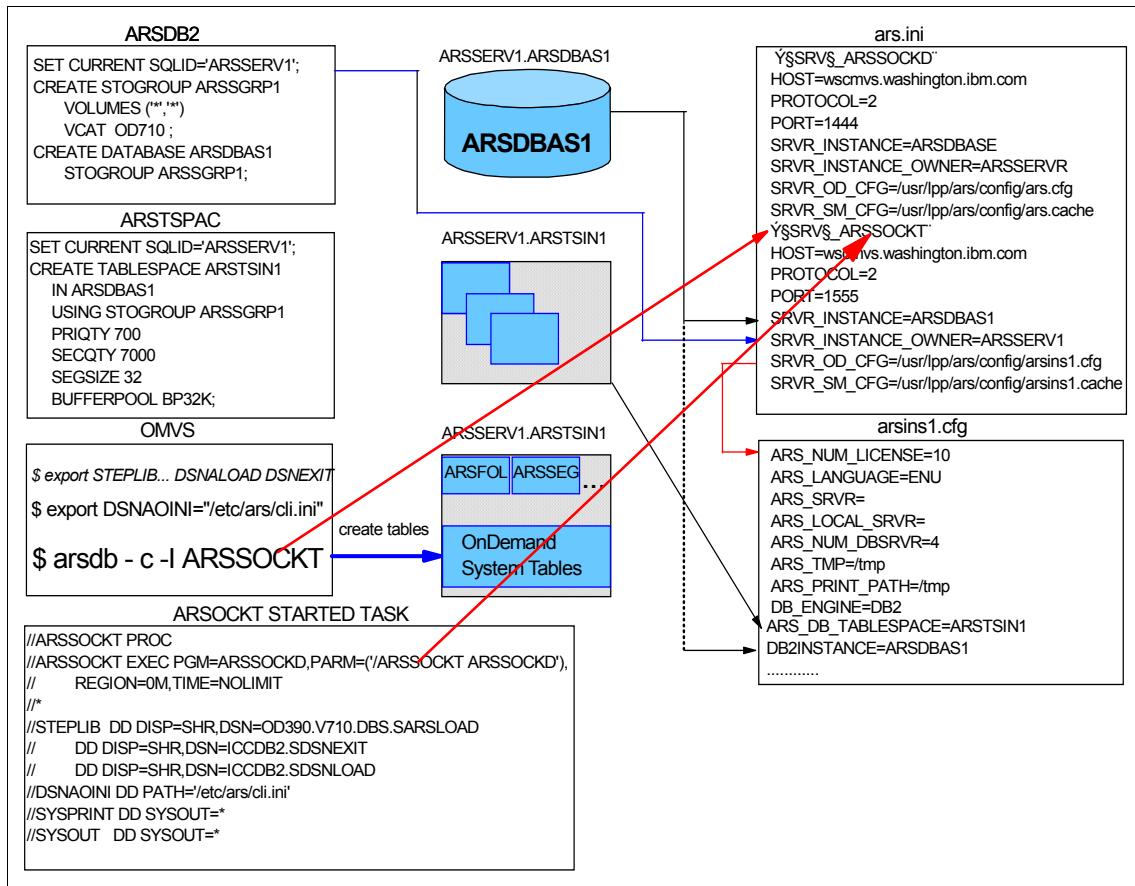


Figure 4-13 Relationship between the steps for creating an instance on z/OS

Running arsload to check the new instance and new file system

After all the configuration work is done and the application group, application, and folder are created, use the **arsload** program for an installation verification. Figure 4-14 shows the procedure used to load data to the new instance. If you see problems in loading the file (writing an object), check the user permissions.

```
//ARSLOAD1 JOB (QFTA0000,B123),
//           'JOHN SMITH', MSGCLASS=0, CLASS=U,
//           NOTIFY=&SYSUID, USER=SYSADM1
//STEP1      EXEC PGM=ARSLOAD, REGION=0M,
// PARM=('/ -u Henry -p xxxxxxxx -n -f -I ARSSOCKET
//           -g American /dev/null')
//*           -I ARSLOCKD -g CKL1 /dev/null')
//STEPLIB    DD DISP=SHR, DSN=OD390.V710.DBS.SARSLOAD
//           DD DISP=SHR, DSN=ICCDB2.SDSNEXIT
//           DD DISP=SHR, DSN=ICCDB2.SDSNLOAD
//           DD DISP=SHR, DSN=OD390.V710.ACIF.V2R2M0.SAPKMOD1
//SYSPRINT   DD SYSOUT=*
//SYSABEND   DD SYSOUT=*
//SYSOUT     DD SYSOUT=*
//INPUT      DD DISP=OLD, DSN=TEAM5.AMERICAN.DATABIN
```

Figure 4-14 ARSLOAD for new instance



Storage management

In this chapter, we explore storage management on various OnDemand platforms. OnDemand uses a cache storage manager for disk storage and supports the use of archive storage managers to keep long-term copies of data on archive storage media. We consider the setup and integration of OnDemand with Archive Storage Manager.

In this chapter, we cover the following topics:

- ▶ Tivoli Storage Manager for Multiplatforms
- ▶ Object access method for z/OS
- ▶ Archive Storage Manager for iSeries

5.1 Tivoli Storage Manager for Multiplatforms

OnDemand for Multiplatforms has a *cache storage manager* that we use to maintain documents on disk storage. The cache storage manager uses a list of file systems to determine the devices that are available for storing and maintaining documents. Typically, each 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 of time to provide the fastest access times for system users. Documents migrate from cache storage to archive storage based on the migration policy that you define for the application group.

OnDemand for Multiplatforms also has an *archive storage manager* that you use to store documents on archive media. The archive storage manager maintains one or more copies of documents and acts as the interface between the object server and archive storage media. Tivoli Storage Manager is included in OnDemand as the archive storage manager. Documents are archived on a variety of media such as disk, optical, and tape. The archive storage devices must be configured and defined to Tivoli Storage Manager.

To store application group data to archive media, you must assign the application group to a storage set that contains a storage node that is managed by the archive storage manager. In an application group definition, you can specify that the data is migrated to archive storage when the document is originally loaded into the system, the next time that the migration maintenance process is run or after a certain number of days pass.

In this section, we consider the steps that you must follow to set up and configure Tivoli Storage Manager as the archive manager for an OnDemand for Multiplatforms system. We discuss configuration of *IBM System Storage Archive Manager*, previously named IBM Tivoli Storage Manager for Data Retention, to store OnDemand data. It provides data retention policies that help meet regulatory requirements and uses storage devices such as IBM TotalStorage DR450, IBM TotalStorage DR550, or EMC Centera.

This section provides an overview with emphasis on the parts of the process that most directly affect the OnDemand to Tivoli Storage Manager interface. It is not meant to provide exhaustive coverage of Tivoli Storage Manager.

5.1.1 Tivoli Storage Manager overview

Before we get started with the installation process, we discuss a few of the components that make up a Tivoli Storage Manager system. For a more complete description of Tivoli Storage Manager, refer to *Tivoli Storage Manager for Windows Administrator's Guide*, GC32-0782.

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

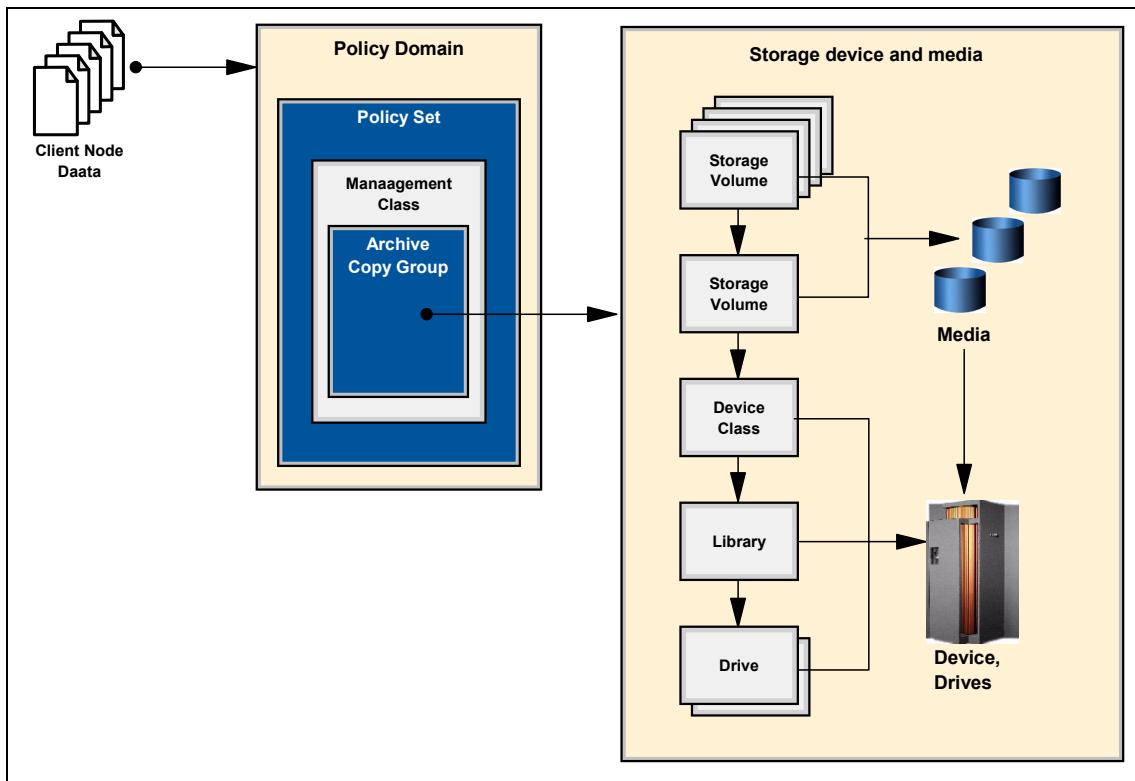


Figure 5-1 Tivoli Storage Manager storage objects

Storage policy

Storage policy consists of the following items:

- ▶ **Client node:** Represents an object server that has the Tivoli Storage Manager backup archive server installed and that has been assigned to a policy domain
- ▶ **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
- ▶ **Drive:** Tivoli Storage Manager-defined drive mechanism in an optical or tape 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

We install and configure Tivoli Storage Manager on a Windows system and then integrate it with OnDemand. While we do not provide comprehensive coverage of Tivoli Storage Manager, we identify areas that are important to the interface with OnDemand. We use Tivoli Storage Manager for Windows Version 5.3 in this discussion.

Refer to *Tivoli Storage Manager for Windows Quick Start*, GC32-0784, which is a good reference to help with installing and configuring the Tivoli Storage Manager system. Within this guide, follow the steps listed for installing 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, continue to the following section that covers the Tivoli Storage Manager configuration.

If you use IBM TotalStorage DR 450 or DR550 for archival, Tivoli Storage Manager is already built into the hardware. No installation is required.

Tivoli Storage Manager configuration

This section covers the standard configuration. There is also a minimal configuration that lets you quickly initialize the Tivoli Storage Manager server and perform a test backup to evaluate the system. We recommend that you use the Tivoli Storage Manager wizard functions to perform the initial configuration tasks.

During the standard configuration process, wizards help you perform the following tasks:

- ▶ Analyze drive performance to determine the best location for the Tivoli Storage Manager server
- ▶ Initialize the Tivoli Storage Manager server
- ▶ Apply the Tivoli Storage Manager licenses
- ▶ Configure Tivoli Storage Manager to access storage devices
- ▶ Prepare media for use by Tivoli Storage Manager

- ▶ Register the client nodes of Tivoli Storage Manager
- ▶ Define schedules for the automation of Tivoli Storage Manager tasks

The standard initial configuration does not cover all Tivoli Storage Manager functions, but results in a functional Tivoli Storage Manager system that can be modified and enhanced further. The default settings used by the wizards are appropriate in many cases.

Initial configuration

After you install Tivoli Storage Manager, perform the following initial configuration:

1. Open the Tivoli Storage Manager management console and expand **Tivoli Storage Manager** until you see the local machine name. Right-click the **local machine name** and select **Add a New TSM Server**
2. From the initial configuration task list, select **Standard** or **Minimal** configuration. Refer to the *Tivoli Storage Manager for Windows Quick Start*, GC32-0784, for information to help you with your decision concerning the configuration type.
3. Select a **Standalone** or **Network** configuration:
 - In a stand-alone environment, a Tivoli Storage Manager server and backup archive client are installed on the same machine. There can be no network-connected Tivoli Storage Manager clients.
 - In a network environment, a Tivoli Storage Manager server is installed. The backup archive client can be optionally installed on the same machine. Network-connected clients can be installed on remote machines.

Performance configuration for Tivoli Storage Manager

After the initial configuration, perform the following for performance configuration:

1. Estimate the number of clients that the Tivoli Storage Manager server supports. Also estimate the size of files to be stored
We recommend that you select the **mostly large files** option, which allows space for files that are generally larger than 1 MB.
2. Tivoli Storage Manager analyzes the local drives to determine the best location for the initial Tivoli Storage Manager database, recovery log, and disk storage pool.

Server initialization

Perform the following actions for server initialization:

1. Choose the directory path for storing files that are unique to the Tivoli Storage Manager server instance.
2. Choose the directories for the Tivoli Storage Manager database, recovery log, and disk storage pool. The analysis performed during the performance configuration results in preferred locations being listed as the default.
3. Choose the logon account and password to be used to start the Tivoli Storage Manager server service and choose whether the service should start automatically at startup or manually be started.
4. Choose an ID and password for the Tivoli Storage Manager server.

After you make the selections for server initialization, Tivoli Storage Manager performs the following actions:

- ▶ Initializes the server database and recovery log
- ▶ Creates the database, recovery log, and disk storage pool initial volumes
- ▶ Creates a daily and weekly schedule that can be used for automated Tivoli Storage Manager functions
- ▶ Registers a local administrative client with the server (The client is named *admin* and the initial password is *admin*.)

License

Select and apply the number of licenses purchased for the different features of Tivoli Storage Manager. The license for Tivoli Storage Manager has been reduced to the following three components:

- ▶ Base IBM Tivoli Storage Manager
- ▶ Base IBM Tivoli Storage Manager Extended Edition
- ▶ IBM Tivoli Storage Manager for Data Retention

Note: If you intend to use the IBM System Storage Archive Manager, then you must obtain and install the license for IBM Tivoli Storage Manager for Data Retention.

The license information provided is registered with the Tivoli Storage Manager server.

Device configuration

The device configuration wizard automatically detects storage devices that are attached to the Tivoli Storage Manager server and is used to select the devices that you want to use with Tivoli Storage Manager. You define a device by selecting the check box that is associated with that device. Undetected or virtual

devices can be manually added. The libraries and drives that you define to Tivoli Storage Manager are available to store data.

Device configuration is not needed if you are using IBM TotalStorage DR550. For EMC Centera, there is no library or drive as well. You must define only the devclass with DEVTYPE=centera to point to the correct IP address.

Client node configuration

Client node configuration allows you to add and register the client nodes to back up the data to the server instance that is being configured. When storage devices were configured during device configuration, storage pools associated with these devices were automatically generated and are displayed here.

Tivoli Storage Manager uses storage pools to represent storage devices. Different storage pools are used to route archive data to different types of storage. Storage pools can be arranged in a hierarchy (Figure 5-2) to allow data to be migrated from one type of storage to another. For instance, you can set up a storage pool hierarchy that stores the data on a hard disk drive, then move to optical media, and finally store the data on tape. The time that the data is stored in each storage pool is determined by the Tivoli Storage Manager policy domain associated with the storage pool.

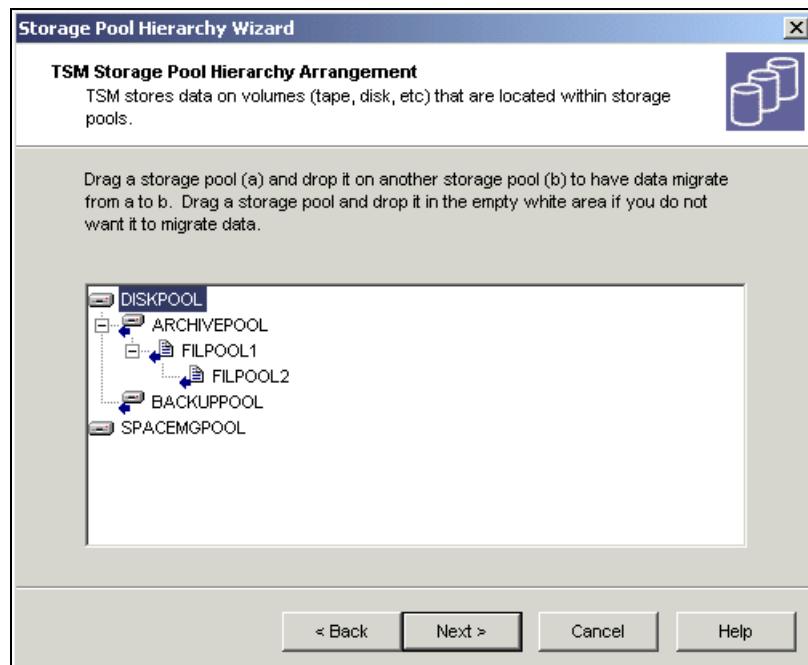


Figure 5-2 Storage pool hierarchy

Tivoli Storage Manager provides a default storage pool named DISKPOOL that represents hard disk drive storage space on the Tivoli Storage Manager server. Three other default storage pools, ARCHIVEPOOL, BACKUPPOOL, and SPACEMGPOOL are also provided. They all point to DISKPOOL.

By default, data that you store with client nodes associated with BACKUPPOOL is transferred to DISKPOOL. The data can be stored in DISKPOOL indefinitely or can be migrated to another storage device in the storage pool hierarchy.

To register new client nodes, you provide a client node name and password for each node that is required (Figure 5-3). The new node defaults to the STANDARD policy domain. BACKUPPOOL is the default storage pool for this policy domain. Associate the client name with the storage pool that is set up to maintain the archive data on the desired device type for the period of time that is required. You can associate the new client node with a different storage pool by selecting New to create a new policy domain.

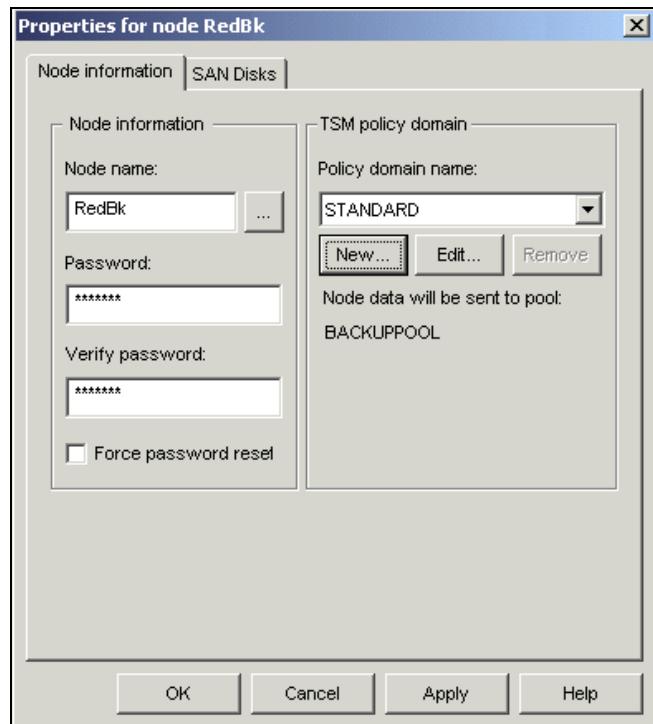


Figure 5-3 Client node configuration

Client nodes that you have registered can be configured to back up data to this Tivoli Storage Manager server instance. The backup data is managed according to the way you set up the client's associated storage pool hierarchy.

Note: The node name and password that you create here are used when creating OnDemand storage sets that uses Tivoli Storage Manager archive.

Client options file

Each client requires a client options file, contains options that identify the node, the server, and the communication method. The client options file, dsm.opt as shown in Example 5-1, can be edited or created using a standard text editor. Refer to *Tivoli Storage Manager for AIX Administrator's Guide*, GC32-0768, or *Tivoli Storage Manager for Windows Administrator's Guide*, GC32-0782, for the format of the options in the file. If the client is for a Windows system, the Network Client Options File Wizard can be used to create the options file.

Example 5-1 The dsm.opt file (for Windows) sample

```
nodename redbk  
  
COMMmethod      TCPip  
TCPPort         1500  
TCPServeraddress 127.0.0.1
```

Verifying the installation and configuration

At this point, you should verify your installation by performing a backup of a file:

1. Start the backup archive client and log on with the node name and password that you created.
2. Click **Backup** from the client window.
3. Expand the directory tree.

4. Select the folder icons and select the boxes next to the files or directories that you want to back up (see Figure 5-4).

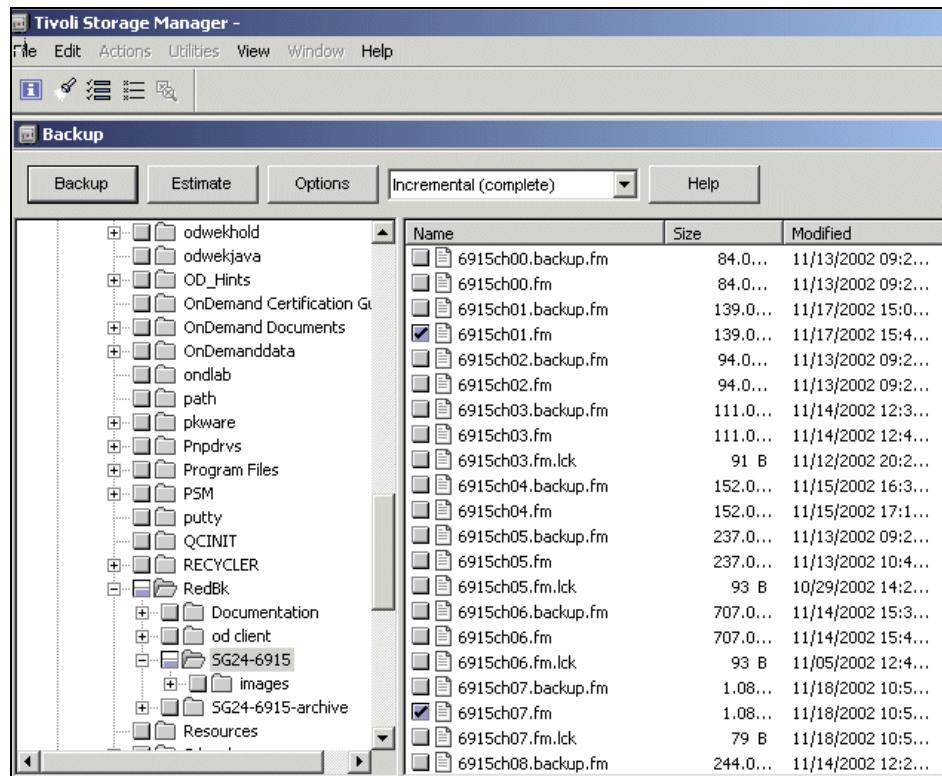


Figure 5-4 Backup archive client

5. From the drop-down list, select the backup type.
6. Click **Backup**. A report window displays the status of the backup.

Note: The first backup of a file is always a full backup, regardless of the backup type that you select.

You can verify that the file was backed up successfully by clicking the Restore option in the backup archive client (Figure 5-5). Then to confirm that the files that you backed up are listed, expand the directory tree under File level. You can also run the restore process to confirm that it is working correctly.

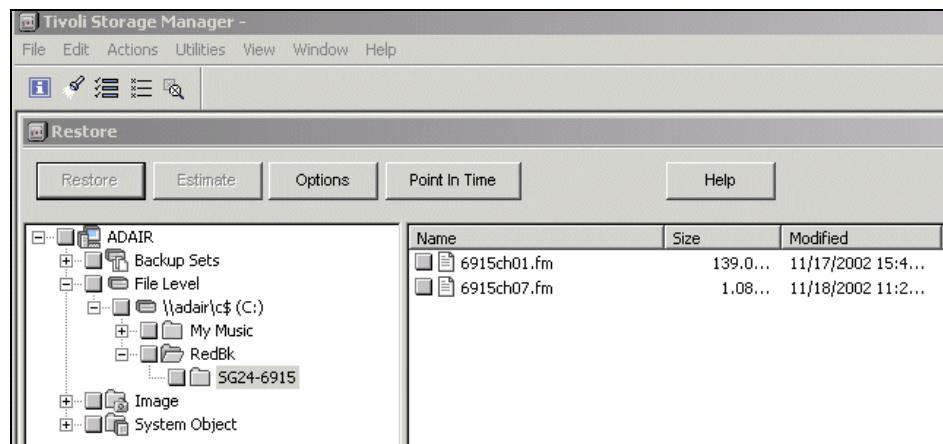


Figure 5-5 Restore client

Note: Tivoli Storage Manager backup and archive client is not supported if data retention protection is turned on. The previous test does not apply if you enabled data retention protection in Tivoli Storage Manager.

5.1.2 Configuring OnDemand for Tivoli Storage Manager archive management

To enable OnDemand to use Tivoli Storage Manager as the archive manager for the system, OnDemand options must be set to allow the system to recognize that Tivoli Storage Manager has been configured for archive storage. In an OnDemand for Windows system, the OnDemand configurator is used to set this parameter. In an OnDemand UNIX-based system, the ars.cfg configuration file is updated to specify that Tivoli Storage Manager is to be used.

In this section, we discuss:

- ▶ OnDemand for Windows Tivoli Storage Manager configuration
- ▶ OnDemand for UNIX Tivoli Storage Manager configuration

OnDemand for Windows Tivoli Storage Manager configuration

If you are configuring an OnDemand for Windows system to use Tivoli Storage Manager for archive storage, the 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 5-6).

You select **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.

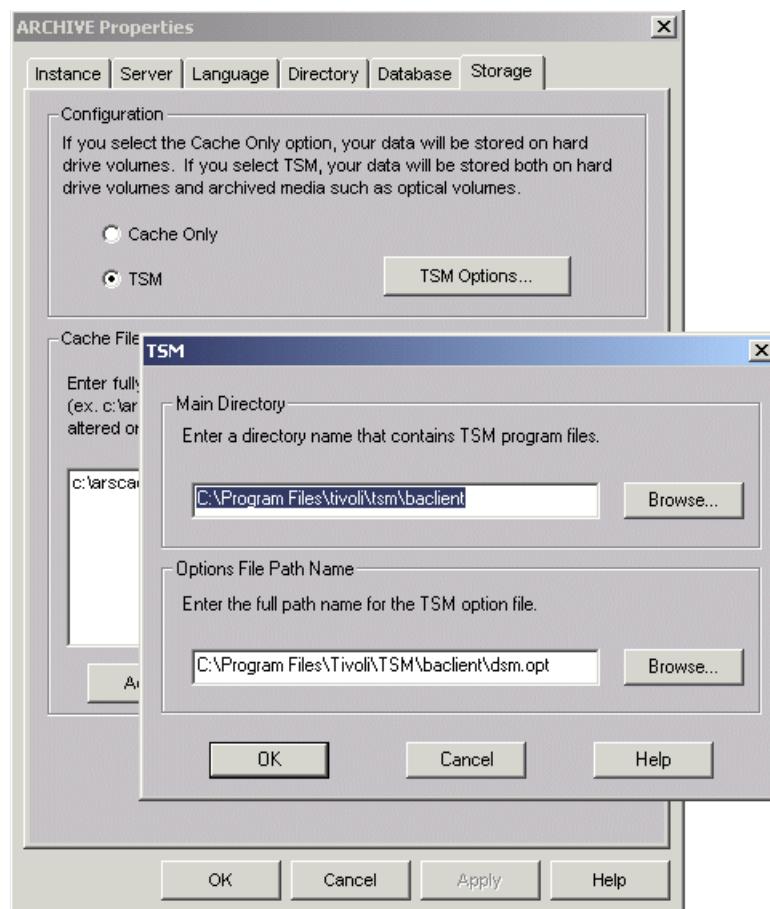


Figure 5-6 Windows configurator

OnDemand for UNIX Tivoli Storage Manager configuration

If you are configuring an OnDemand for UNIX system to use Tivoli Storage Manager for archive storage, you must be sure that the ars.cfg file (Figure 5-7) has been updated to reflect that Tivoli Storage Manager is to be used as the storage manager. The file must also include valid paths for Tivoli Storage Manager options files and all of the Tivoli Storage Manager components to be used.

```
#####
# 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/ba/bin
DSM_CONFIG=/usr/tivoli/tsm/client/ba/bin/dsm.opt
DSM_LOG=/ondemand/arslog
DSMG_DIR=/usr/tivoli/tsm/client/api/bin
DSMG_CONFIG=/usr/tivoli/tsm/client/api/bin/dsm.opt
DSMG_LOG=/tmp
DSMI_DIR=/usr/tivoli/tsm/client/api/bin
DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin/dsm.opt
DSMI_LOG=/tmp
```

Figure 5-7 ARS.CFG Tivoli Storage Manager configuration

Note: For the Tivoli Storage Manager client used by OnDemand, we recommend that you set COMPRESSION NO in the Tivoli Storage Manager client option file, dsm.opt for Windows or dsm.sys for AIX. Because OnDemand objects are compressed before they are sent to Tivoli Storage Manager for archival, compression by Tivoli Storage Manager is not required.

5.1.3 OnDemand storage management

The storage management criteria that you specify on the OnDemand library server determines where and when OnDemand stores reports and how those reports are maintained. Figure 5-8 illustrates OnDemand storage object relationships. When a report is loaded into 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.

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 hard disk drive storage. Another can be set up to point to a Tivoli Storage Manager client node that will cause a copy of the report to be stored in archive storage.

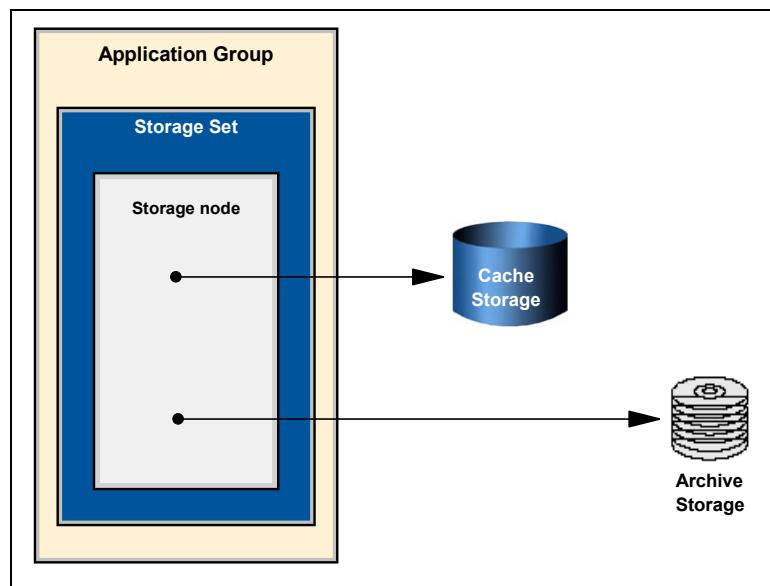


Figure 5-8 OnDemand storage objects

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

Note: The date that is used to determine the *Life of Data and Indexes* in OnDemand is the date field index value taken from the report that is being loaded. The date used for the *retention period* in Tivoli Storage Manager is the date that the report is first migrated to Tivoli Storage Manager.

If the load type value for the application group is *load*, a command is issued from OnDemand to Tivoli Storage Manager to delete data when the data is being expired from OnDemand. If the load type is *segment* or *document*, a delete command is not issued from OnDemand to Tivoli Storage Manager when 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 OnDemand because the indexes are expired in OnDemand.

5.1.4 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 stored in an application group. A storage node is associated with a specific OnDemand object server. When Tivoli Storage Manager is used for archive storage, each storage node associated with Tivoli Storage Manager-managed storage 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.

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 using Tivoli Storage Manager to maintain and manager the archive of OnDemand documents and indexes over a predesignated period of time. When 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 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 OnDemand administrator defines and maintains storage sets (Figure 5-9). The load type is the storage set parameter that we examine here.

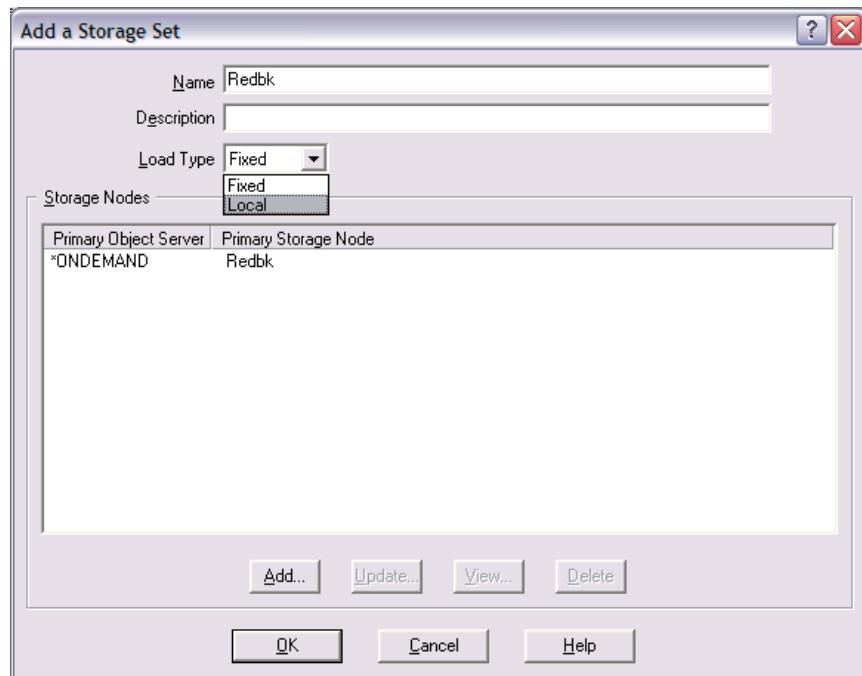


Figure 5-9 Storage set definition

Load Type

The Load Type parameter determines where OnDemand stores data. There are two possible values (Figure 5-9):

- ▶ **Fixed:** 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. OnDemand loads data to only one primary storage node regardless of the number of primary nodes that are defined in the storage set.
- ▶ **Local:** OnDemand stores data in a primary storage node on the server on which the data loading program executes. When 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 reside on one or more object servers.

On the primary node panel (Figure 5-10), there are several parameters that we must examine.

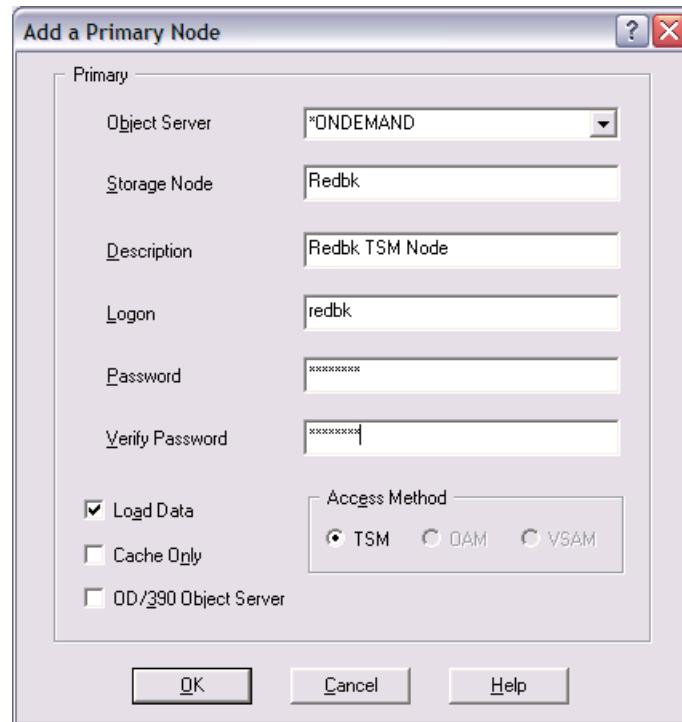


Figure 5-10 Primary node definition

Storage Node

The OnDemand storage node name can be from one to sixty characters in length and can include embedded blanks. The case can be mixed.

OnDemand no longer supports adding secondary storage nodes when you create a storage set.

Note: The 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 OnDemand system. The storage node name can be the same as the associated client node name, but it is not required that they are the same.

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 is the client node that has been 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. OnDemand uses a 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 OnDemand loads data. When the load type is fixed, one primary storage node must have load data selected. 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 OnDemand uses the archive manager for long-term storage of data.

After we install and configure Tivoli Storage Manager, create an OnDemand storage set, and assign it to a Tivoli Storage Manager client node, we are ready to consider how an application group uses the cache storage manager. We must also consider how the archive storage manager is to store, maintain, and expire OnDemand report data.

5.1.5 Application group storage management

The application group storage management settings (Figure 5-11) determine how long report data and indexes are kept in cache storage before being expired. There are also choices to be made concerning how soon data is migrated to the archive storage after data is loaded.

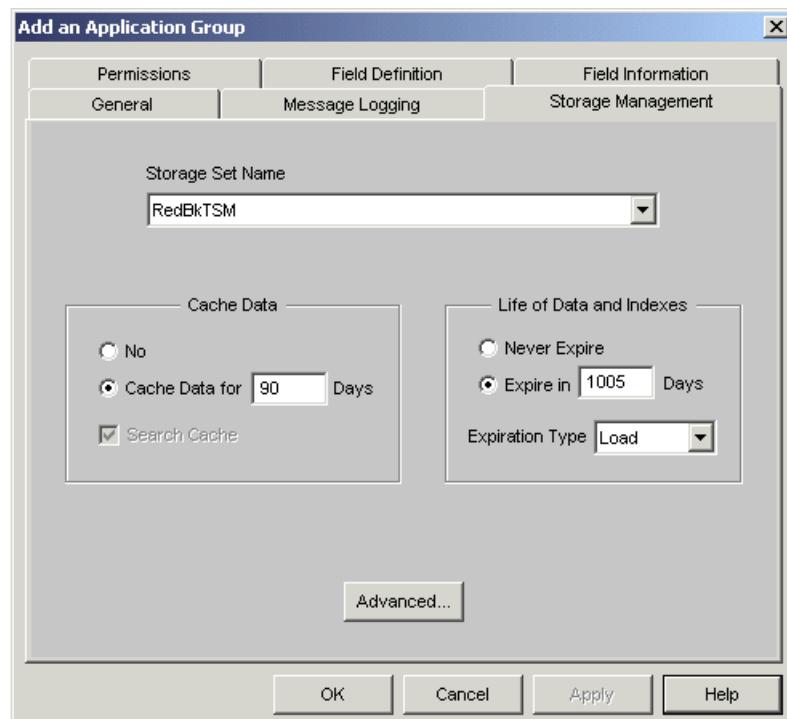


Figure 5-11 Application group storage management

Cache Data

The Cache Data setting determines if 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 retrieving 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 should generally remain in cache until it is no longer needed by 90% of 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 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 has been reached, the **arsmaint** command can be used to expire the data from the system.

Expiration Type

The Expiration Type determines how report data, indexes and resources are expired. There are three expiration types:

- ▶ **Load:** With this expiration type, 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 OnDemand deletes the data. OnDemand signals to the storage manager that the data might be deleted. Load is the recommended expiration type.
- ▶ **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 have been 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 of time and the data is not be 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 with an expiration type of document causes the expiration process to search through every document in the segment to determine if the expiration date has been reached resulting in long processing times.

When the **arsmaint** expiration process is run, data is only deleted from the application group if the upper threshold for the size of cache storage has been reached. By default, the cache threshold is 80%. A lower threshold can be forced by the expiration command parameters. Unless there is some reason that cache must be cleared, leaving data in cache improves retrieval performance.

5.1.6 Advanced application group storage management

The advance storage management settings (Figure 5-12) allow you to adjust the size of the load object and to determine when report data, indexes, and resources are migrated to archive storage.

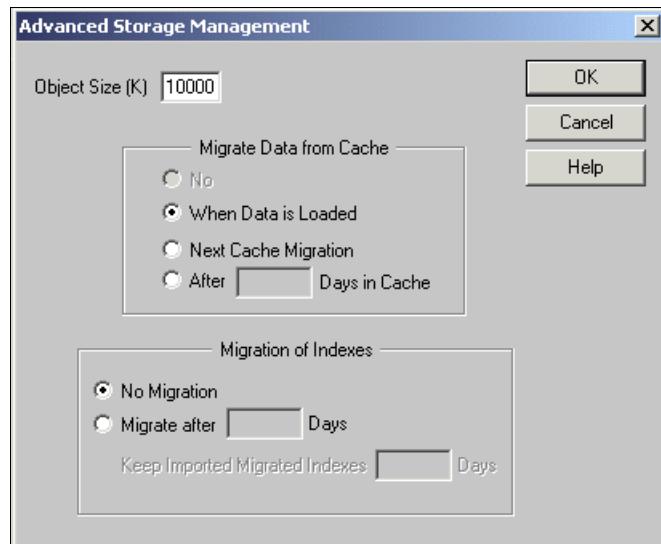


Figure 5-12 Advanced application group storage management

Object Size

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

Attention: Use care when changing the value for Object Size. Setting the value too small or too large can have an adverse affect on load performance.

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

Migrate Data from Cache

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

- ▶ **No:** Data is never migrated from cache. This option is unavailable when a storage set 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 to be copied from cache to archive storage.
- ▶ **After __ days in cache:** This value specifies the number of days that data is to remain in cache only storage. After reaching the prescribed number of days in cache storage, the data is copied to archive storage the next time that ARSMAINT is run with the -m option for data migration.

5.1.7 OnDemand with IBM System Storage Archive Manager

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

Because disk storage has become more affordable over the years and with technology advancement, disk storage devices that prevent data from being erased or overwritten have started to become popular. These WORM disks can be used to store data just as the WORM tapes or optical platters. IBM System Storage Archive Manager allows critical data to be retained for a mandated period of time without the possibility of being rewritten or erased.

In this section, we discuss the enhancements in OnDemand that use the WORM disk. We also provide some setup recommendations and pointers when configuring OnDemand to use such devices.

IBM System Storage Archive Manager

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

Previously known as *IBM Tivoli Storage Manager for Data Retention*, this feature was available in Tivoli Storage Manager 5.2.2. It is used to prevent critical data from being erased or rewritten. For more information about the IBM System Storage Archive Manager, refer to the following Web address:

<http://www.ibm.com/software/tivoli/products/storage-mgr-data-reten/>

IBM System Storage Archive Manager provides new functions and new device 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 OnDemand on loads, unloads, application groups deletes, and expiration of data.
- ▶ **Event-based retention policy:** Data is retained based on a time interval after the occurrence of a retention-initiating event. For OnDemand, this is a call to delete the data. A load, 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. OnDemand does not take advantage of this feature.
- ▶ **New device support:** Support is available for all the devices (more than 400 storage devices) that Tivoli Storage Manager Extended Edition supports.

OnDemand operation with Tivoli Storage Manager server API

With the new event-based retention policy, the object expiration can now be event based instead of just creation-based. There is a new option in the archive copygroup definition called the RETINIT. It determines the time when the retention time specified by the RETVER attribute is initiated. There are two possible values:

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

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

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

Table 5-1 Comparison of Tivoli Storage Manager expiration methods with data protection OFF or ON

DRP	TSM RETinit	OnDemand action: Unload	OnDemand action: Delete Application group
OFF	Creation	<p>The Delete Object command is issued through the Tivoli Storage Manager API.</p> <p>Objects are deleted during the next Tivoli Storage Manager expiration.</p>	<p>The Delete Filespace command is issued.</p> <p>Objects are immediately deleted along with the file space.</p>
	Event	<p>OnDemand issues an event trigger command through the Tivoli Storage Manager API.</p> <p>The status of the objects 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 will never expire.</p>	<p>The Delete Filespace command is issued.</p> <p>Objects are immediately deleted along with the file space.</p>
ON	Creation	<p>OnDemand issues no commands to Tivoli Storage Manager.</p> <p>The objects are effectively orphaned by OnDemand and are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects will never expire.</p>	<p>OnDemand issues no commands to Tivoli Storage Manager.</p> <p>The objects are effectively orphaned by OnDemand and are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects will never expire.</p>
	Event	<p>OnDemand issues an event trigger command through Tivoli Storage Manager API.</p> <p>The status of the objects 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 will never expire.</p>	<p>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 the affected objects is changed from PENDING to STARTED. They are expired by Tivoli Storage Manager based on their retention parameters.</p> <p>This 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.</p>

OnDemand 8.3.1 (7.1.2.1) setup recommendations

The following recommendations are applicable to OnDemand V8.3.1 (also known as V7.1.2.1) and later:

- ▶ Application groups should be set up to expire by load.
- ▶ Tivoli Storage Manager archive copy groups should be defined to be event-based and retain data 0 days.
- ▶ Tivoli Storage Manager inventory expiration should be run regularly to ensure expired data is cleaned up.

Configuring data protection with IBM TotalStorage DR550

The IBM TotalStorage DR550 is an integrated, preconfigured, complete hardware and software offering. It is designed to help store, retrieve, manage, share, and secure regulated and non-regulated data in a non-erasable and non-rewritable format. It helps customers protect the integrity of their data. IBM System Storage Archive Manager is used as the control code that manages the IBM TotalStorage DR550.

In the DR550, the Tivoli Storage Manager database, database volumes, recovery log, recovery log volumes and primary storage pools and storage pool volumes are all preconfigured. You are not required to define anything for DR550 if you use the default setting.

The DISK device class is used by primary storage pool called ARCHIVEPOOL. There is also a DBBKUP device class with device type FILE that is used for database backup.

You can attach a tape device for the purpose of backing up your primary storage pools to copy storage pools. You can also use the tape device to back up the Tivoli Storage Manager database. Tape devices are well-suited for this, because the media can be transported off-site for disaster recovery purposes. A tape drive or tape library is not included in the IBM TotalStorage DR550/DR550 Express. However, you can attach tape devices that are supported by Tivoli Storage Manager on the AIX platform and that best suit your data retention requirements.

Since Tivoli Storage Manager is already built-in, configuration is simple. The OnDemand library or the object server is defined as a client node on Tivoli Storage Manager server. If you do not want to use the default domain, policy set, management class and archive copygroup, you can either modify it or create new set of definition on the Tivoli Storage Manager server. On the OnDemand server, which is the Tivoli Storage Manager client, set the dsm.opt and dsm.sys file to point to the correct IP address with the defined node name.

Remember to define the dsm.sys file with the following option:

```
ENABLEARCHIVERETENTIONPROTECTION YES
```

The client node has the same option as a normal OnDemand client.

Note: Remember to set ARCHDELETE=yes for the client node on the Tivoli Storage Manager server. If this is not set, you will experience errors when you try to delete the application groups or unload data from OnDemand.

For more information about DR550, refer to the IBM Redbooks publication *IBM System Storage DR550 Setup and Implementation*, SG24-7091.

Configuring data protection with Centera

EMC Centera is a disk-based system and is treated as a device by Tivoli Storage Manager. The Tivoli Storage Manager server must be running data retention protection. Centera devices can also be used as a standard storage device if no mandatory retention requirements exist for the data.

For use with Centera, the Tivoli Storage Manager database must be a new database that has not previously stored any data. Nor should any data have been previously loaded onto the server.

Configure the Tivoli Storage Manager server as normal; however, you are not required to define a library or drive for the Centera storage device. To define devclass, use the new command DEFINE DEVCLASS CENTERA.

To enable Centera support for data retention protection, use the new command on the Tivoli Storage Manager server:

```
SET ARCHIVERETENTION PROTECTION
```

In the dsm.sys file, specify in the following option:

```
ENABLEARCHIVERETENTIONPROTECTION YES
```

Note: Similar to IBM TotalStorage DR550, you should set ARCHDELETE=YES for the node client that is used for OnDemand.

Operations that are not supported with Centera

Certain server operations are not supported if the device class associated with the storage pool that has a device type of *Centera*. As such, there is no copy storage pool configured, which is handled by Centera storage.

The following operations are not supported with Centera:

- ▶ Migration
- ▶ Reclamation
- ▶ Moving node data into or out of a Centera storage pool
- ▶ Backing up Centera storage pools
- ▶ Restoring Centera storage pool volumes
- ▶ Exporting data to a Centera device class or importing data from a Centera device class

Files stored in Centera storage pools can be exported, and files being imported can be stored on Centera.

- ▶ Using a Centera device class for creating backup sets
Files stored in Centera storage pools can be sent to backup sets.
- ▶ Defining Centera volumes
- ▶ Using a Centera device class to back up a database
- ▶ Using a Centera device class for database loading or unloading
- ▶ Using a Centera device class as the target of volume history, device configuration, trace logs, error logs, or query output files

Note: The data stored in Centera devices cannot be moved anymore.

For more information about Tivoli Storage Manager support of Centera devices, see the *Tivoli Storage Manager for AIX Administrator's Guide*, GC32-0768, or *Tivoli Storage Manager for Windows Administrator's Guide*, GC32-0782.

5.1.8 The **arsmaint** command

We have referenced the 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 OnDemand database and in cache storage. It maintains the system using the storage management values that are specified for application groups. It is typically run in 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 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 expiring cache data, by default, the data is not expired until the cache storage file system has exceeded 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 cache is 80% full by using the minimum and maximum parameters to override the percentage full default.

Refer to *IBM Content Manager OnDemand for Multiplatforms - Administration Guide*, SC18-9237, for a detailed explanation of the **arsmain** command and its associated parameters, along with all other OnDemand commands.

5.2 Object access method for z/OS

In this section, we provide an introduction to object access method (OAM) and show its relationship with OnDemand in a z/OS environment. For more information about setting up OAM, refer to 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 the IBM Redbooks publication *Image and Workflow Library: Content Manager for ImagePlus on OS/390 Implementation and EIP*, SG24-4055

OAM is the DFSMSdfp™ component that manages a class of data, called *objects*, in a z/OS environment. *Objects* are bit strings that are handled as one big byte string rather than processing them as records, as is done with data sets. The content of this byte string is not known to OAM. There are no restrictions 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 is designed to handle an unlimited number of objects, which can be stored on magnetic disk, magnetic tape, or optical storage. Objects are different from data sets, which are handled by existing access methods. The following characteristics distinguish them from traditional data sets:

- ▶ **Lack of record orientation:** There is no concept of individual records within an object.
- ▶ **Broad range of size:** An object might contain less than one KB or up to 50 MB of data.

- ▶ **Volume:** Objects are usually much smaller than data sets; however, they can use much more external storage, depending on the kind of application creating 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.

5.2.1 OAM components and SMS terminology

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

OAM components

The functions of OAM are performed by three components:

- ▶ Object Storage and Retrieval (OSR) component

This component provides an API for OAM. All OAM API functions are requested via the OSREQ assembler macro. Applications use this interface to store, retrieve, query, and delete objects, as well as 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 invoked through the application programming interface 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 reside. 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 objects should be stored 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 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 some 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 names of the storage classes are set up by the storage administrator based on the naming convention in the Enterprise. These storage classes are:

- ▶ **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 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, along with storage classes, help reduce the requirement for users to understand the physical characteristics of the storage devices which 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:

- ▶ 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 data sets. Class attributes control backup, migration, retention of data, and release of unused space. OSMC uses information from the management classes to determine which automatic management processes should be performed upon corresponding OAM objects.

Automated Class Selection routine

Automated Class Selection (ACS) routines are used to assign class and storage group definitions to data sets and objects. ACS routines are written in the ACS language, which is a high-level programming language that is similar to that used for the construction of TSO CLISTs. The ACS translator is used to convert the routines to object form so 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 a large number of objects, which, if cataloged separately, can require an extremely 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 one and only one Object storage group. Each storage group can contain from one to many collections.

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

5.2.2 Defining a storage set

When the OnDemand administrator defines a new storage set, the Add a Storage Set window opens as shown in Figure 5-13.

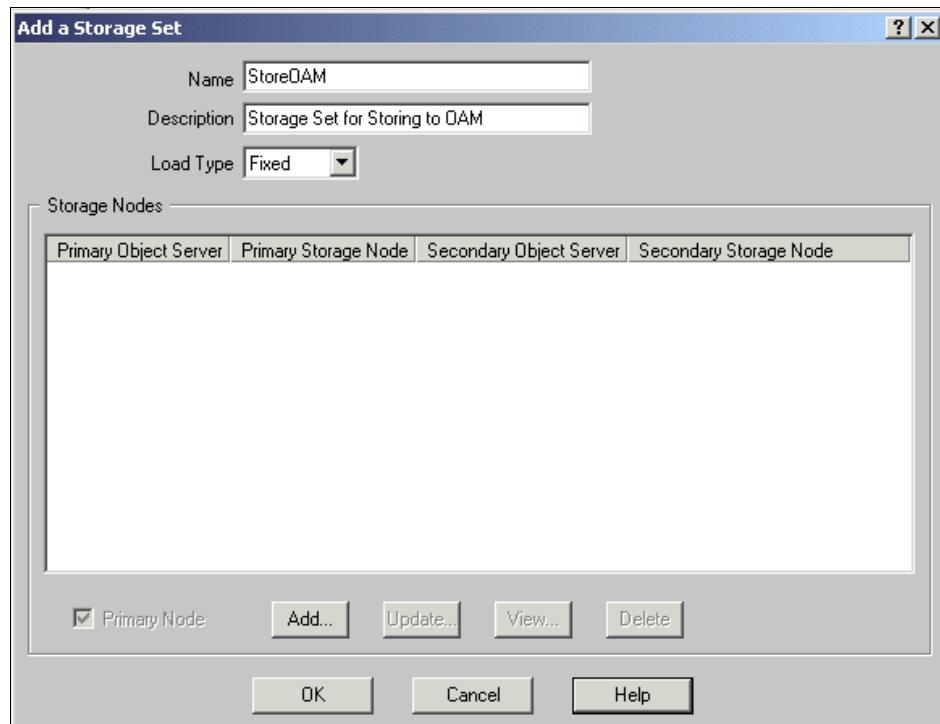


Figure 5-13 Adding a storage set

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

- ▶ **Name:** The name of the storage set
- ▶ **Description:** The storage set description, up to 120 characters
- ▶ **Load Type:** Where OnDemand stores data

There are two choices:

- *Fixed:* 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. There can be several different collection names.
- *Local:* OnDemand stores data in a primary node on the server on which the data loading program executes. This applies to z/OS.

Then the administrator clicks **Add** to add a primary storage node to this storage set. Then the Add a Primary Node window opens as shown in Figure 5-14.

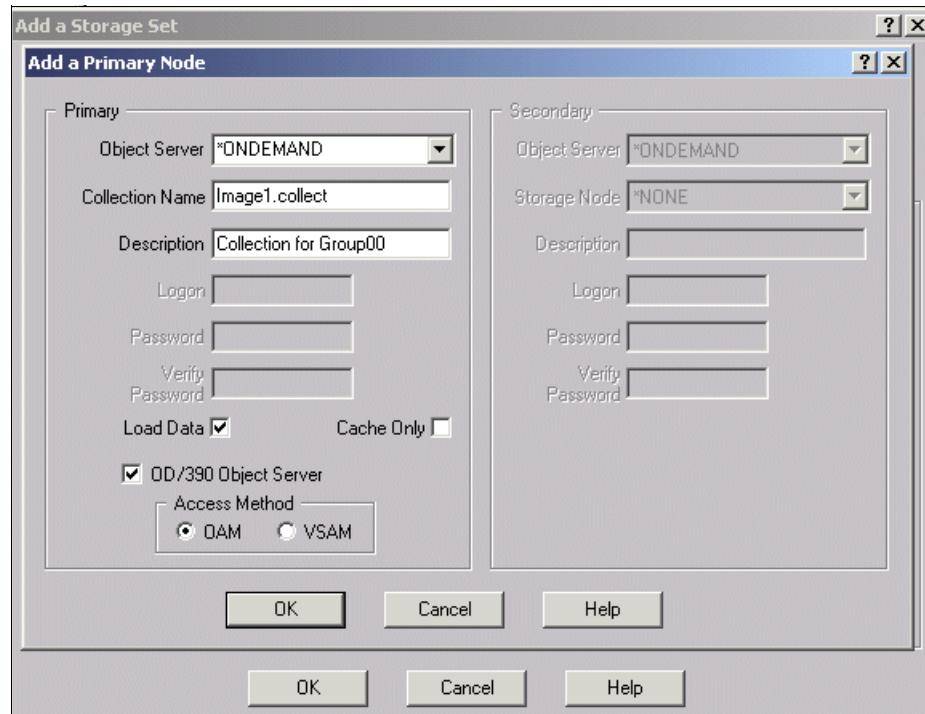


Figure 5-14 Adding a primary storage node to the storage set

The object server is always OnDemand if the OD/390 Object Server check box is selected. The *load data* check box indicates that the data is loaded to this collection. You must select the OAM check box. The Logon and Password fields are not used in a z/OS environment. These fields are for Tivoli Storage Manager only.

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

Figure 5-15 shows the relationship between the creation of storage sets and OAM.

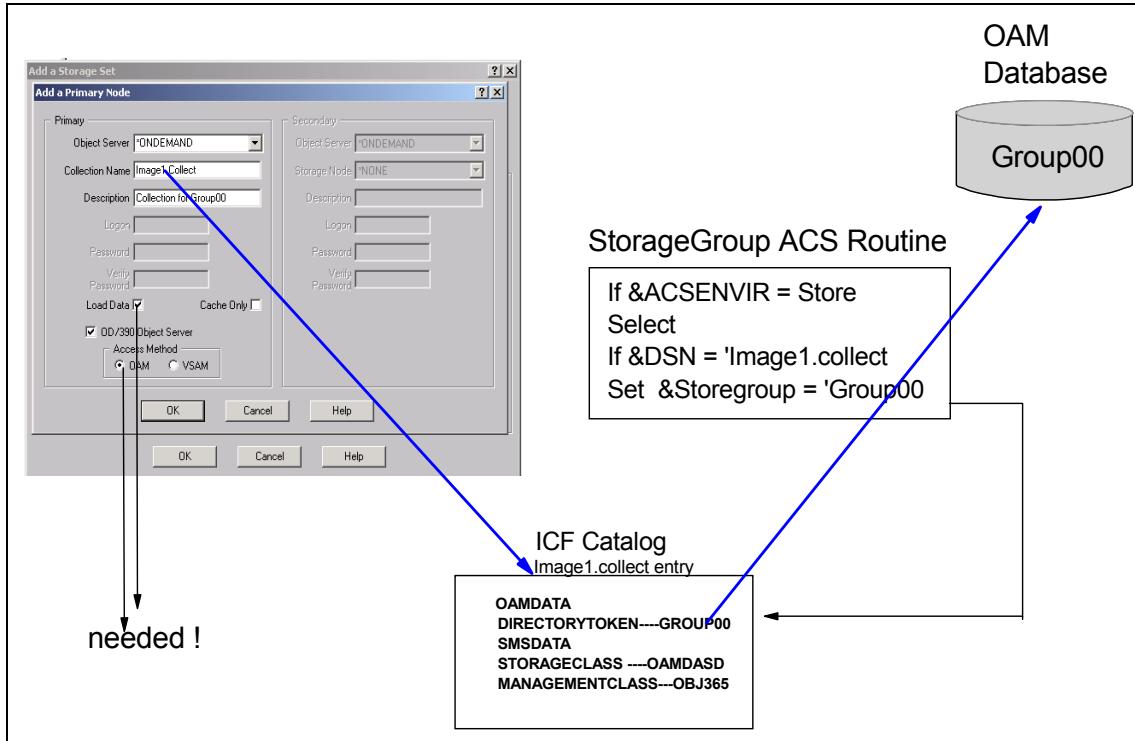


Figure 5-15 Relationship between OAM and OnDemand

Object naming conventions

The object name identifies the object within a collection. The object name is unique within a collection and is provided by the OnDemand application. Currently no installation exits allow for any customization of these names. The object name is composed of the application group name, the load identifier within the application group portion of the load ID. The load identifier within the application group is composed by 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 this:

A AAAAAAA.L1.FAAA

The maximum size of an object is specified via the OnDemand Administration GUI when defining an application group. The default value is 10 MB. Currently, the maximum size for an OAM object is 50 MB. The OnDemand administrator must be careful not to specify a value exceeding this limit.

Attention: In the current implementation, OnDemand is not aware that an object has been deleted by OAM based on management class criteria set by the Storage Management component. A user can search for data which is no longer available. There is no synchronization between OAM object expiration and index expiration. Be sure to define the index expiration correctly when defining the application group.

Figure 5-16 shows the window in which you can set up the expiration for Storage Management when defining or updating an application group.

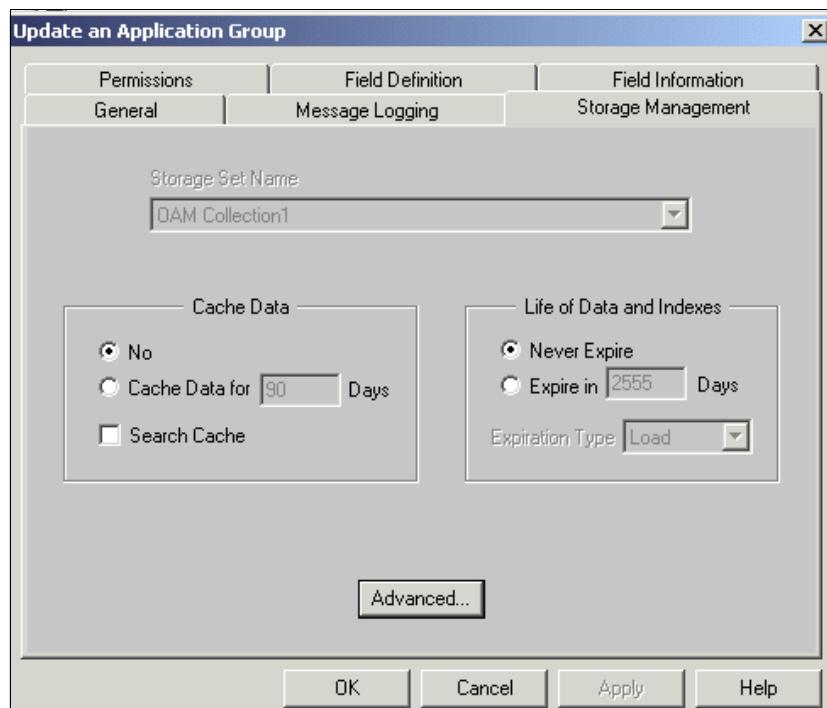


Figure 5-16 Defining index expiration in OnDemand

Tip: OnDemand and OAM can run in different DB2 subsystems (different DB2 subsystem identifiers (SSIDs)).

5.2.3 Storing data to Virtual Storage Access Method data sets

Another way to store data on the z/OS system is via the Virtual Storage Access Method (VSAM). OnDemand can create objects that are stored in VSAM data sets. All storage management issues for VSAM data sets such as allocation, backup, and migration apply for these object data sets.

To create a storage set that stores to VSAM, the OnDemand administrator must provide the first level qualifier for the defined cluster statement. In the example shown in Figure 5-17, TEAM5 is the high (first) level qualifier.

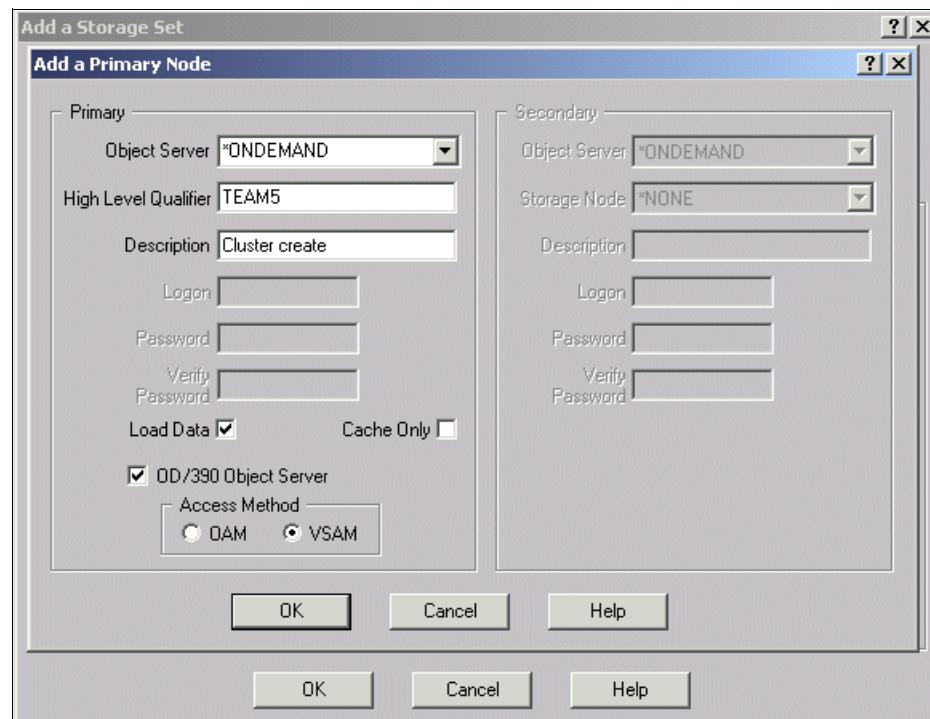


Figure 5-17 Defining a storage set for VSAM

Based on these parameters, OnDemand creates VSAM data sets during the `arsload` program. A catalog entry is created as shown in Example 5-2.

Example 5-2 VSAM data set name

TEAM5.FAA.L1.FAAA

This is done automatically by the OnDemand system. The only part that you can create for yourself is the first level qualifier. The space allocation during the

Define Cluster is done by the OnDemand code as well. The default object size set when defining the application group influences the number of bytes for the primary and the secondary allocation. The number of bytes is divided by 16 for the primary allocation. Every time an **arsload** is done with this storage set, this amount of data is allocated even if the objects are much smaller.

Every load creates two VSAM data sets, one for the data, and one for the index. Every Define Cluster of a VSAM data set is a catalog entry. If you have several million loads with this storage set, your catalog can grow very large.

You can browse the VSAM data set; 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 data set is viewable. Compression can be switched off on the load information on the application panel.

If you store Advanced Function Presentation (AFP) data to VSAM, the resources are stored in a different VSAM data set.

5.3 Archive Storage Manager for iSeries

OnDemand for iSeries Disk Storage Manager maintains a copy of documents on disk. Disk Storage Manager migrates documents from cache to the Archive Storage Manager. Archive Storage Manager then migrates documents to archive media.

Archive Storage Manager maintains one or more copies of documents on archive media, such as disk pool, optical or tape. The OnDemand administrator decides which type of media that the OnDemand system requires, configures the storage devices on the systems, and defines the storage devices to Archive Storage Manager. To store application group data on archive media, the application group must be assigned to a storage set that is managed by Archive Storage Manager.

When creating an application group, the OnDemand administrator specifies how long documents should be maintained on the system and whether the index data should be migrated from the database to archive media. OnDemand system management programs use this information to migrate documents from disk to Archive Storage Manager, delete documents from disk, migrate index data from the database to archive media, and delete index data from the database. OnDemand can then reclaim the space that had been used by the migrated and expired data.

Disk Storage Manager indicates to Archive Storage Manager when to expire data based on the Life of Data and Indexes, under Application Group → Storage Management. Archive Storage Manager deletes data from the archive media when it reaches its storage expiration date. The OnDemand administrator defines management information to the archive storage manager for the OnDemand data that is to be managed. This management information includes storage volumes that can contain OnDemand data, the number of copies of a report to maintain, and the amount of time to keep data in the archive management system.

5.3.1 Migration policy

Migration policies and storage sets must be defined before you can define reports to OnDemand or load data into the system. Migration policies contain migration and storage media characteristics for data archived using OnDemand. The information is used by Archive Storage Manager to determine if and when archived data should be moved 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 OnDemand Common Server. This change was made to the program product code because of performance problems that customers encountered when archiving a large amount of data and leaving it in the Cache directory. Even though document retrieval is fast, the load process takes longer as the size of the cache directory grows.

Also, the Cache Only storage set was limiting because you could not add any storage levels to it. Even when this storage set was automatically created at installation, many customers chose to define a disk pool and create a migration policy instead. Then if they later decide to begin using an optical library, they can easily add an optical storage level to the policy.

If you have been using the Cache Only storage set, you may decide to start using a migration policy instead for greater flexibility and to avoid archival performance problems. The OnDemand Administrator Client does not allow you to change the storage set in the application group.

However, here are three different ways to make the change from Cache Only to a migration policy:

- ▶ Rename each application group and application (for example, add a suffix of OLD to the names). Copy the application group and application to the original names, and change the storage set in the application group to the newly created migration policy.

From that point on, documents are archived using the migration policy. Documents already archived remain in Cache. This technique might be acceptable if you do not have a large amount of data or do not keep the archives for a long time. This technique is also the easiest and most foolproof change to make. However, it cannot be used with application groups migrated from Spool File Archive. If you rename migrated application groups, the data can no longer be retrieved.

- ▶ Rename the application groups and applications and create new ones as described previously. Then re-spool the documents and archive them again.

One way to do this is to retrieve a list of *all* the documents within a folder, select them all, and print them to a server printer. The server printer should point to an output queue that does not have an active writer. Then the output queue can be monitored and all documents archived into OnDemand. For each spooled file created, a field such as userdata or formtype must be modified to match the application group and application name so that the output queue monitor can be used to automatically archive the files.

You must be careful and make sure that you reprint and re-archive all the data. When you finish the entire process, you can delete the original application groups, which also delete all the data archived in those groups.

- ▶ Rename and create new application groups and applications as described earlier. Use the **arsdoc get** API in the Qshell environment to retrieve the compressed data, indexes, and resources for each archived file. This information can be created in an integrated file system directory.

Then use the **arsload** command to archive the data into the new application groups. This technique can be used by customers who are familiar with the Qshell environment. Again, you must be careful to retrieve and re-archive all the data before deleting the renamed application groups.

When you create a migration policy, a storage set of the same name is automatically created by OnDemand. If you plan to keep all your archives on disk, the best approach is to create a disk pool and a migration policy that specifies “No Maximum” for the duration level. Archive Storage Manager expires data and indexes whenever the number of days is reached in the Life of Data and Indexes in the application group, *or* whenever an expiration level in the migration policy is encountered, whichever comes first. If there is *no* expiration level in the migration policy, data is only expired according to the Life of Data in the

application group. If you plan to add an optical level later, you can specify 90 days, for example, for the ASP01 disk pool level, with no other storage levels. When an optical level is added later, the archives are moved from the disk pool level to the optical level. With this technique, make sure that you *never* add an expiration level after the disk pool level because, if that level is encountered, the archives will be expired.

In the QPRLCASM1 status report created by Archive Storage Manager, you might see messages indicating that the number of days in the ASP01 level has been exceeded since there is no level 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, then a copy of the data is archived to the integrated file system CACHE directory and to the integrated file system ASMREQUEST directory. When you run Disk Storage Manager, the data is deleted from cache after the Cache Data for Days duration has passed. When you run Archive Storage Manager for the first time after loading data, the data is moved to the first level of the migration policy, ASP01 in our example. 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 an OnDemand for iSeries Common Server can be carried out with the OnDemand administrator client. Creating the objects necessary for OnDemand archive storage management on the iSeries must be done through iSeries Access Navigator with the OnDemand plug-in (Figure 5-18 on page 145).

To create a migration policy, there must be storage devices defined for the types of archive media required by the OnDemand system. For the purposes of our scenario, we created a disk pool storage group and an optical storage group.

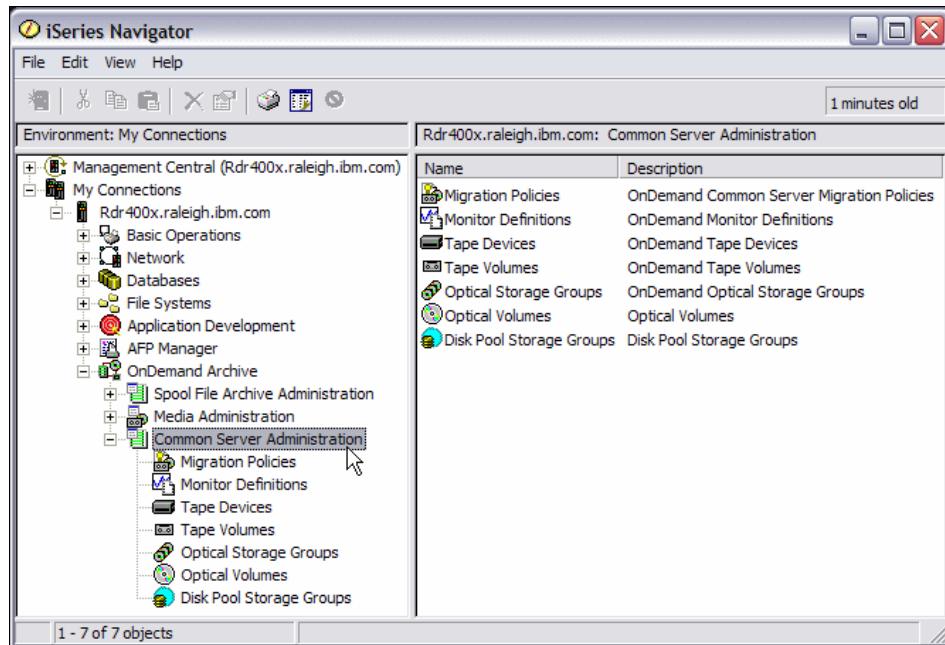


Figure 5-18 iSeries Access Navigator

Disk pool storage group

A disk pool storage group is used to identify an OS/400 auxiliary storage pool that Archive Storage Manager uses as disk storage media when migrating archived data. Use iSeries Navigator to add a disk pool storage group (Figure 5-19).

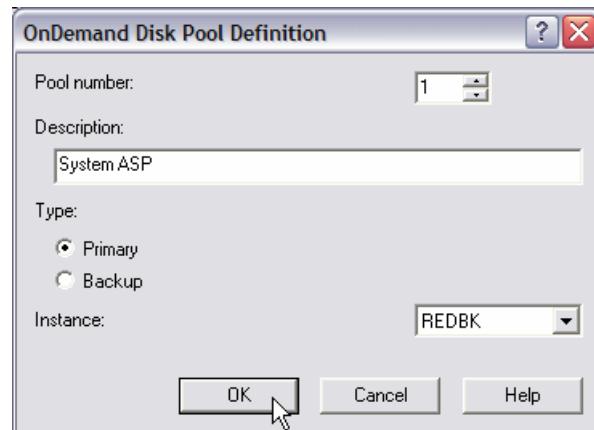


Figure 5-19 iSeries Disk Pool definition

Provide the following information for Disk Pool definition (Figure 5-19 on page 145):

- ▶ A pool number that corresponds to an existing auxiliary storage pool
- ▶ A description of the storage group
- ▶ The type of data, primary or backup
- ▶ The OnDemand instance with which the storage group is associated

Optical storage group

Optical storage groups are used by OnDemand to group sets of optical volumes for the storage of related data. By using a specific storage group in the migration policy, the administrator can control which sets of reports are stored on specific optical volumes. Use iSeries Navigator to define the optical storage group (Figure 5-20).

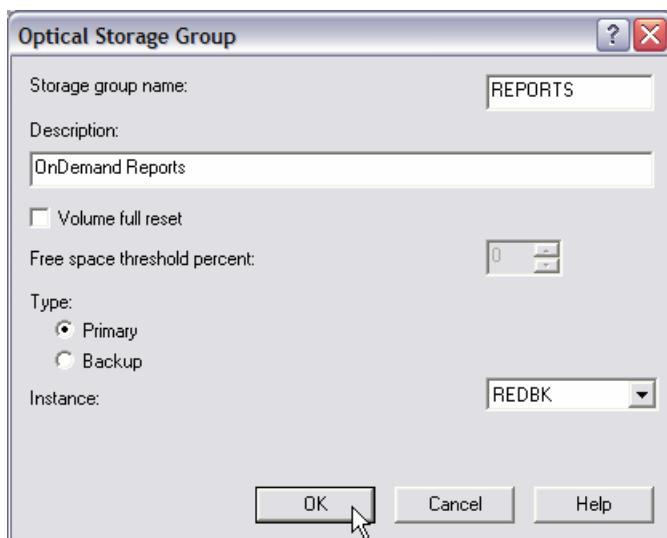


Figure 5-20 iSeries optical storage group

When defining the optical storage group (Figure 5-20), you provide:

- ▶ 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 LAN-attached optical jukeboxes)
- ▶ Free space threshold percent (the percent at which OnDemand starts storing to rewritable volumes again if the volume full reset parameter is checked)
- ▶ Storage group type, primary or backup

- The OnDemand instance with which the storage group is associated

After you define the optical storage group, use iSeries Navigator to define the optical volumes to the OnDemand system (Figure 5-21).

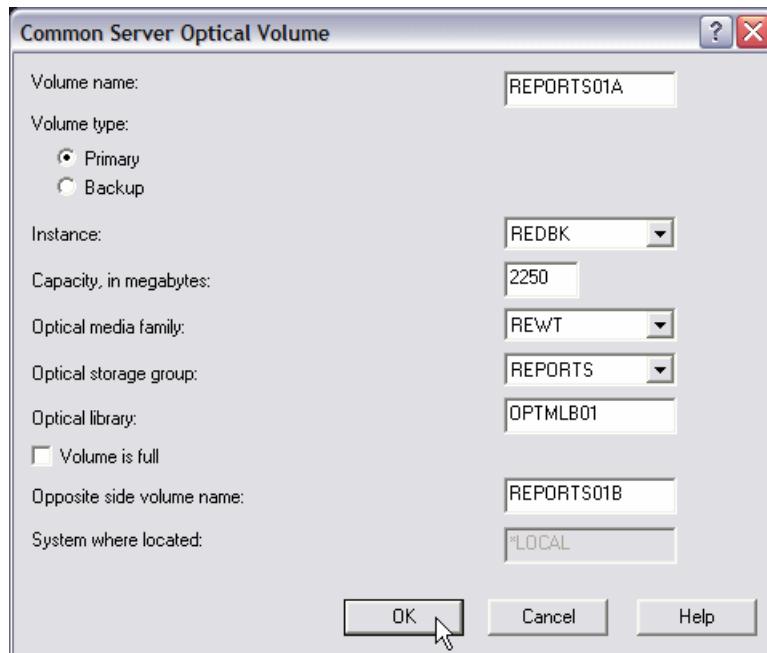


Figure 5-21 iSeries optical volume

When defining optical volumes (Figure 5-21), you provide this information:

- **Volume name:** Your volume name
- **Volume type:** Primary or backup
- **Instance:** OnDemand instance with which the optical volume is associated
- **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) used by DVD RAM drives, or Universal Disk Format double-sided (UDF2)
- **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

After the storage groups are established, use iSeries Navigator to define the migration policy needed to use the storage groups (Figure 5-22).

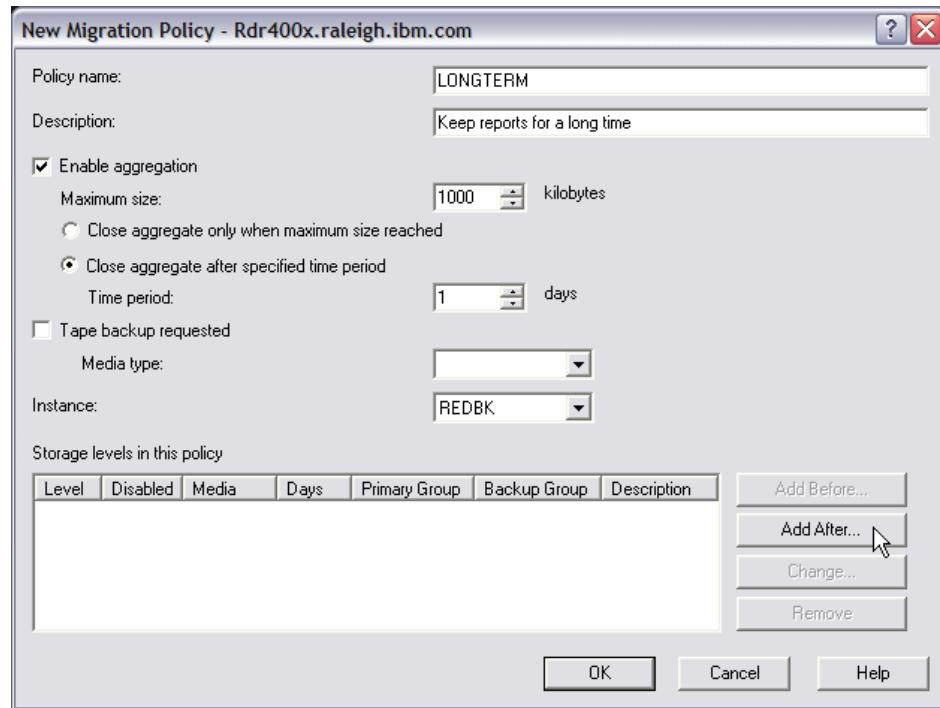


Figure 5-22 iSeries migration policy

The migration policy definition (Figure 5-22) includes:

- ▶ **Policy name and description:** This field is for the policy name and its description.

Tip: It is a good practice to put information such as “length of time and where located” in the *description* rather than in the *policy name* field. This is because you can change, add, and delete levels, but you cannot change the name. You do not want to have a name that is no longer accurate.
- ▶ **Enable aggregation:** If selected, Archive Storage Manager combines individual archived objects into larger objects to provide a more efficient process. Archive 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. Archive Storage Manager closes the existing aggregate and opens a new aggregate when the maximum value is reached.
- ▶ **Close aggregate only when maximum size reached:** If selected, the aggregate stays open until the maximum is reached.
- ▶ **Close aggregate after specified time period:** This value in this field specifies the number of days before an aggregate closes. Archive Storage Manager 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 prior to the migration of the object from disk to the first Archive Storage Manager storage level. Only aggregate files that have closed are eligible for migration by Archive Storage Manager. 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 OnDemand objects might remain on disk longer than the period specified by the application group. Choosing to close the aggregate after a specified period of time addresses this problem.

- ▶ **Tape backup requested and media type:** The Tape backup requested field indicates whether a one-time tape backup should be made of the data before it is archived. The Media type field indicates the type of tape to use for the backup.
- ▶ **Instance:** The value in this field indicates the OnDemand instance with which the migration policy is associated.
- ▶ **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 the Add Before or Add After button. The New Policy Level window (Figure 5-23 on page 150) opens.

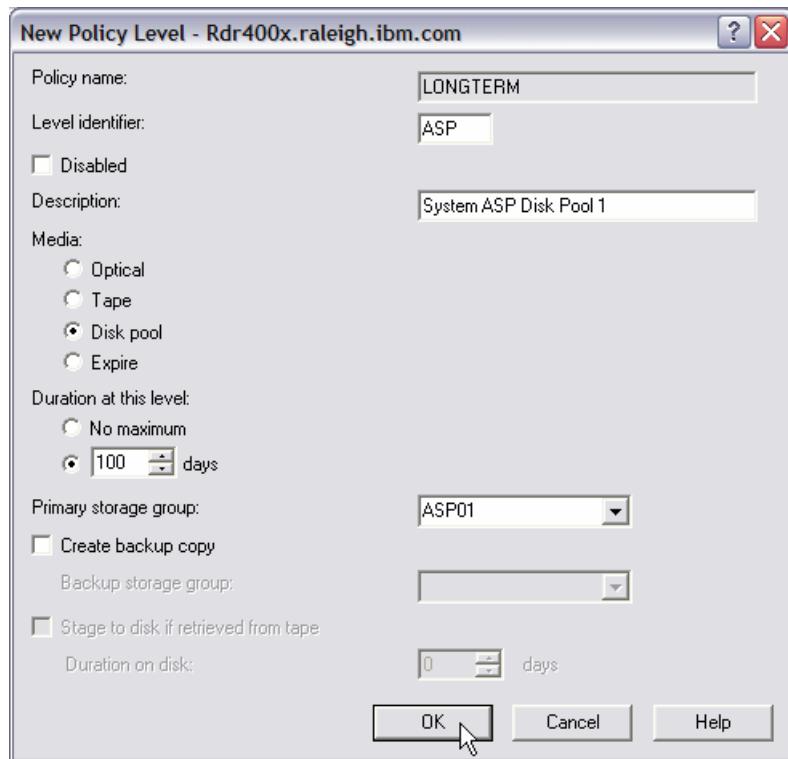


Figure 5-23 iSeries new policy level

In the New Policy Level window, you provide the following information for the new storage level (Figure 5-23):

- ▶ **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. Archive Storage Manager uses the level identifier to determine current level of the migration hierarchy and to determine the next level to which the data should be moved. The identifier can be numeric (for example, 10, 20, and 30) or descriptive (for example, ASP or OPT).
- ▶ **Disabled:** Specifying option causes Archive Storage Manager 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 to be discontinued, such as a tape unit. A policy level might not be removed if data is archived to it, but it might be disabled so that no more data gets migrated to that level.
- ▶ **Description of the policy level:** Use this field to provide a description of the policy level.

- ▶ **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 specified in the application group has not been exceeded. It is not necessary to specify an expire level. Instead, you can let the data expire when it has exceeded the number of days 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 Archive Storage Manager 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 Archive Storage Manager to create a backup copy of the data when it moves to this policy level. The backup storage group must have been created with a type of backup.
- ▶ **Stage to disk if retrieved from tape and duration on disk:** Choose these options to cache data returned from tape to disk for the number of days specified.

In our scenario, we created a policy level that stores data for 100 days on disk using the disk pool storage group assigned to auxiliary storage pool 1. We also created a policy level that stores data on optical indefinitely and uses the REDBK optical storage group. We did not include an expire level, so the data will always be expired 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 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 5-24 on page 152 shows the final migration policy structure.

Note: When the migration policy is created, a corresponding storage set is created for the OnDemand instance with which the migration policy is associated. The storage set is displayed in a listing of storage sets using the OnDemand Administrator Client but can only be viewed. No updates can be made to existing storage sets, and no new storage sets can be added using the Administrator Client. Storage sets in the OnDemand for iSeries system can only be created and modified through the use of iSeries Navigator and migration policies.

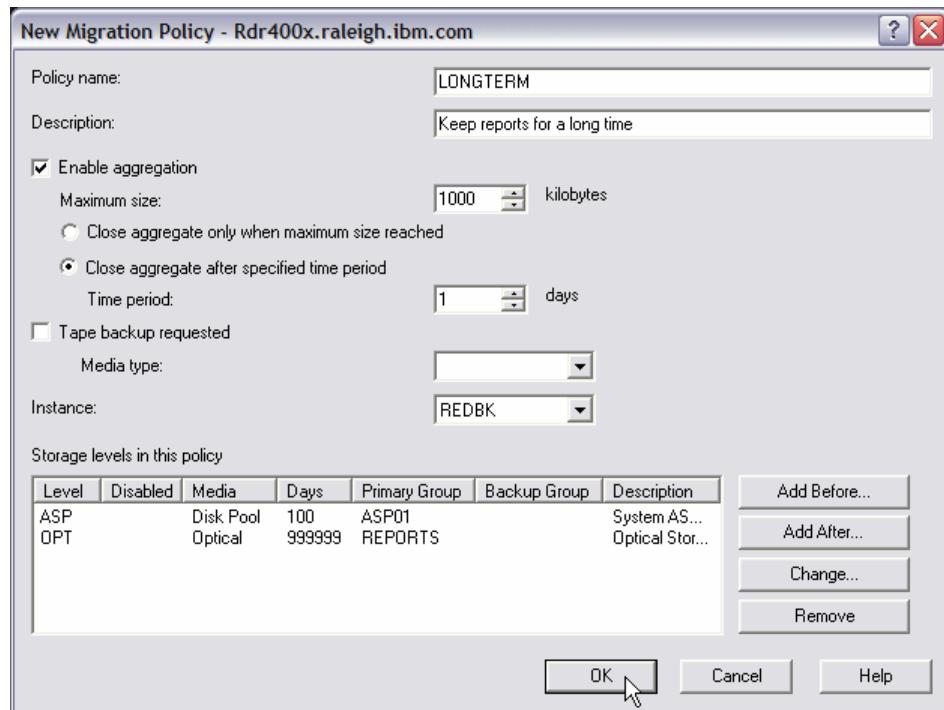


Figure 5-24 iSeries migration policy hierarchy

5.3.2 Application group storage management

The application group storage management settings (Figure 5-25 on page 153) 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 defining the application group, choices are made concerning how soon data is migrated to archive storage after the report load is completed.

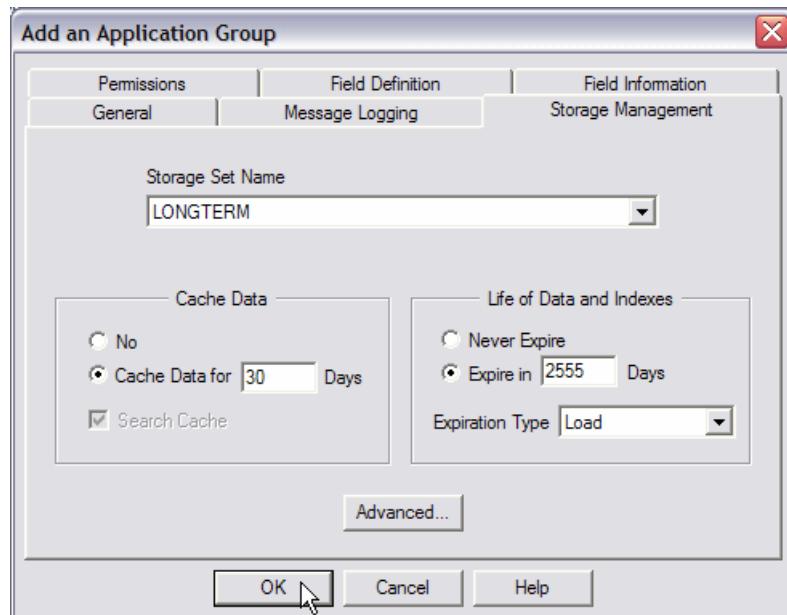


Figure 5-25 iSeries application group storage management

Cache Data

The Cache Data setting determines if the report data is stored in disk cache, and if so, how long it is kept in cache before it is expired. If the Cache Data for n Days option is selected, then the search cache is always selected.

Search cache determines whether OnDemand searches cache storage when users retrieve documents from the application group. When you set Cache Data to No, you can configure 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 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.

Note: 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 have exceeded the cache data or life of data periods. Refer to *IBM Content Manager OnDemand for iSeries Common Server - Administration Guide*, SC27-1161, for details about running the STRDSMOND command.

Expiration Type

The Expiration Type determines how report data, indexes, and resources are expired. There are three expiration types:

- ▶ 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 OnDemand will delete 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 have been 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 of time 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 if the expiration date has been reached resulting in long processing times.

Note: Expiration Type of *Load* is not allowed when using the `arsdoc add` API or when using the workstation APIs such as those used by the OnDemand Toolbox Store Component (see 15.4, “OnDemandToolbox” on page 496). If you plan to use these APIs with an application group, specify the Expiration Type as *Document*.

5.3.3 Advanced application group storage management

The advanced storage management settings (Figure 5-26) allow you to adjust the size of the load object and to determine when report data, indexes, and resources are migrated to archive storage.

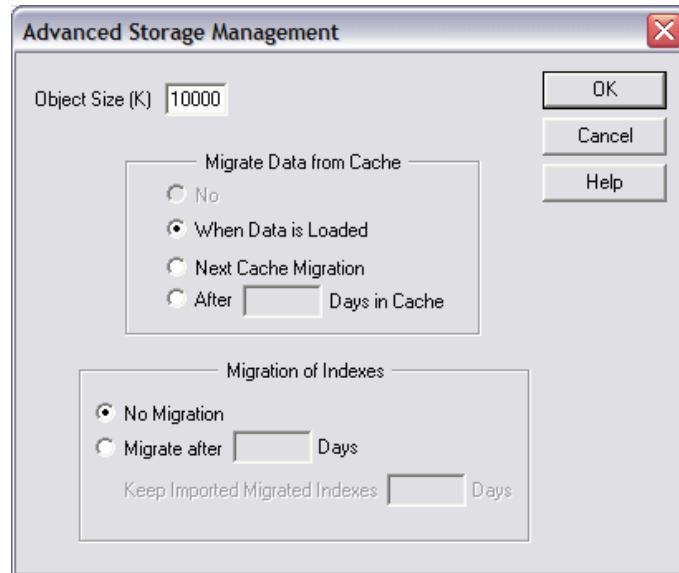


Figure 5-26 iSeries application group advanced storage management

Object Size

The Object Size parameter determines the size of a storage object in kilobytes. OnDemand, by default, segments and compresses stored data into 10 MB storage objects. The default 10 MB is the recommended object size value.

Attention: Setting the value too small or too large can have an adverse affect on load performance.

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

Migrate Data from Cache

This section of the Advanced Storage Management window determines when documents and resources are migrated to archive storage. A storage set associated with a migration policy using archive media must be selected to enable migration to archive storage. The possible values are:

- ▶ **No:** Data is never migrated from cache. This option is unavailable when a storage set 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 from one of the store commands such as Add Report to OnDemand (ADDRPTOND), STRMONOND, or **arsload**.
- ▶ **Next Cache Migration:** Data is migrated to archive storage the next time that Archive Storage Manager is run or when Disk Storage Manager is started with the ASM(*YES) parameter.
- ▶ **After __ Days in Cache:** This value specifies the number of days that data is to remain in cache only storage. After reaching the prescribed number of days in cache storage, the data is copied to archive storage the next time that Archive Storage Manager is run or if Disk Storage Manager is run with the ASM(*YES) parameter.

Note: The archive storage manager is started with the STRASMOND command. The command should only be run in batch. Refer to *IBM Content Manager OnDemand for iSeries Common Server - Administration Guide*, SC27-1161, for details concerning running the STRASMOND command.

5.3.4 Advanced application group database information

The default value for the size of the database (index records) for an application group is 10000000 records. When the database file reaches that number, another one is automatically created.

Customers who have a very large number of database records for an application group might choose to specify the *Single table for all loads* option in the Advanced panel of the Application General Information panel (see Figure 5-27). This way, there is no maximum to the number of database records kept in a single file for an application group.

However, if you use the Save Changed Objects (SAVCHGOBJ) command when doing daily backups, you might prefer to keep the default database size. You only save the most recent file instead of always saving one large file.

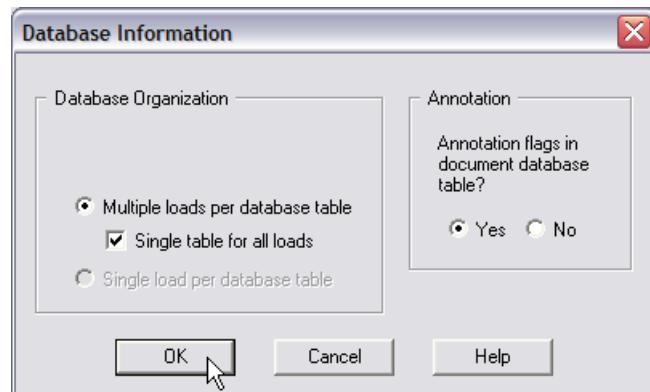


Figure 5-27 Single table for all loads



Performance

In this chapter, we discuss the ways in which the various components within 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. We provide performance guides for the server and OnDemand Web Enablement Kit (ODWEK) components and consider issues related to specific data types such as Portable Document Format (PDF).

In this chapter, we cover the following topics:

- ▶ When performance tuning is necessary
- ▶ Tuning OnDemand to enhance performance
- ▶ Performance issues based on data type

6.1 When performance tuning is necessary

Due to the way in which OnDemand integrates with a wide variety of products, data types, and operating systems spanning over many different vendors, it should come as no surprise to learn that a standard installation of OnDemand is not optimized for all of these different environments. As part of the installation process, decisions must be made and parameter values must be changed from the default settings to best configure the product to fit the environment on which it must operate. In many cases, it might not be possible to anticipate future demand and workload which will be placed on the system. Therefore, as requirements change over time, it might be necessary to fine-tune the system to maintain high level of performance.

An example of an area where performance tuning is sometimes required is the data loading process. The load process, which is followed when the **arsload** program is executed, is shown in Figure 6-1. This process diagram should help you to determine which part of the process might be in need of performance tuning. You can see from the diagram that, within the indexing phase of the overall loading process, only two key components can affect performance:

- ▶ The indexing parameters defined by an OnDemand administrator
- ▶ The report file

If you are experiencing poor performance during indexing, it is likely that one of these two areas is the cause of the problem.

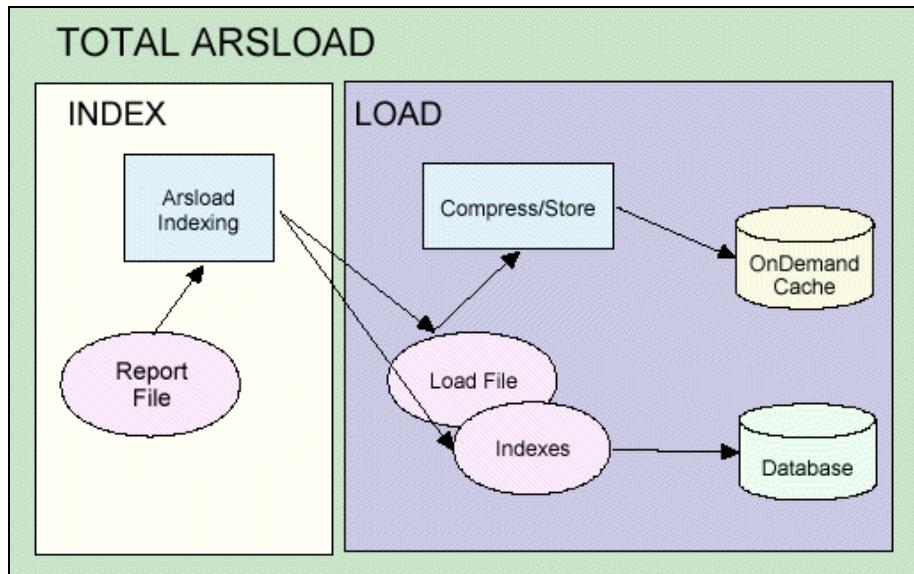


Figure 6-1 The *arsload* process

Figure 6-1 on page 160 also shows a high level view of the actual data loading phase of the overall load process. In this phase, both the database and the data storage areas (such as the cache and optical volumes) ingest the indexes and the data. Poor performance in the loading phase is likely to be caused by problems at either the database manager or the storage manager level.

To help you troubleshoot any performance problems that you might be experiencing with OnDemand, we provide a list in Table 6-1 of the most common areas, where performance tuning might be necessary, that are cross-referenced to the sections within this chapter where you can find help and guidance.

Table 6-1 Performance troubleshooting reference

Problem area	Section to reference
Loading data	► 6.2.4, “System management” on page 166
► Indexing	► 6.2.2, “OnDemand configuration” on page 163 ► 6.3, “Performance issues based on data type” on page 177
► Storing	► 6.2.5, “Storage management” on page 169 ► 6.2.3, “Database” on page 164
Retrieving data	See the following subsections:
► Searching for a document	► 6.2.3, “Database” on page 164 ► 6.2.2, “OnDemand configuration” on page 163
► Viewing a document	► 6.2.2, “OnDemand configuration” on page 163 ► 6.2.6, “ODWEK configuration” on page 170 ► 6.3, “Performance issues based on data type” on page 177
Logon to OnDemand	► 6.2.2, “OnDemand configuration” on page 163 ► 6.2.3, “Database” on page 164

Notice that Table 6-1 specifically covers OnDemand operations and processes that might require tuning to enhance performance. Other areas, such as the underlying operating system, hardware including network, or even contention with other software running on the same machine, might require tuning for better performance. However, in this chapter, we only deal with the areas to tune within OnDemand; it is by no means a definitive reference for all performance problems experienced on the machine where the OnDemand system is installed.

6.2 Tuning OnDemand to enhance performance

The following sections describe the various components of an OnDemand system and architecture in turn and provide guidance about parameters and configurations that you can change to improve performance under certain circumstances. For troubleshooting a specific problem that you might be experiencing, consult Table 6-1 on page 161 for the corresponding section in this chapter.

6.2.1 OnDemand architecture

From a performance tuning perspective, one of the most significant features of the OnDemand product is its architecture. As illustrated in Figure 6-2, there are four significant aspects of the OnDemand architecture:

- ▶ **Library server:** For logon, report definitions and searching the database
- ▶ **Object server:** For storage management of data
- ▶ **Client:** Can be ODWEK or a Windows thick client
- ▶ **Network:** Connecting these components together

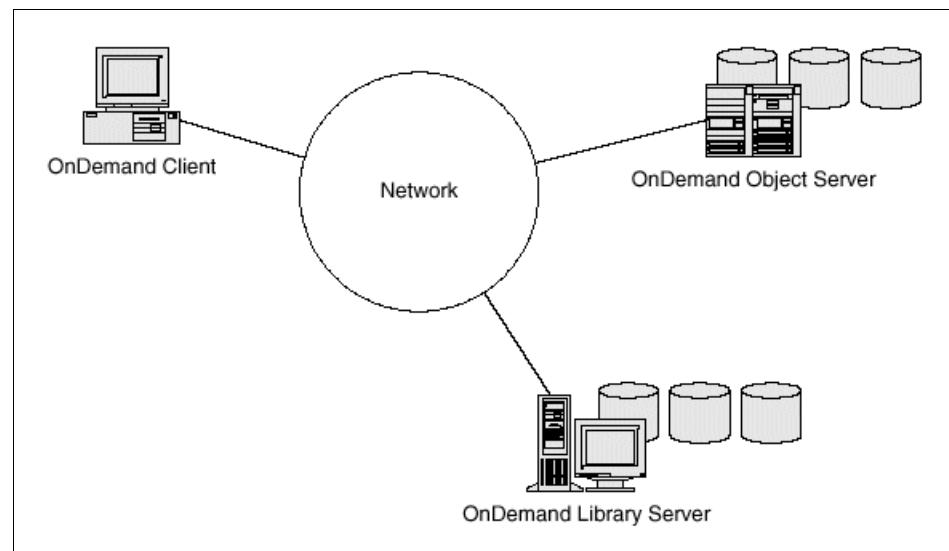


Figure 6-2 The OnDemand architecture

The design of this architecture is based on performance considerations. The ability to separate the object server from the library server delivers 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

In practical terms, when an OnDemand architecture has distributed object servers, it means that the load job (**arsload**) is physically executed on the object server where you intend to store the reports. As the **arsload** program runs, the indexing is done on the object server and the reports are loaded to the local storage manager. Meanwhile the indexes that are generated are sent to the library server to be loaded into the database. A common method of significantly increasing the performance of loading data into OnDemand is to have two object servers running the load program simultaneously. For more information about running parallel load jobs, see “Running parallel **arsload** jobafp2pdfs” on page 167.

In addition to the performance enhancements in the load process, there can also be improvements in document retrieval times if the object servers are distributed nationally or internationally. A common configuration for an OnDemand system that must span large geographical areas is to place an object server in each of the main computer centers. With this configuration, users can search their documents from the central library server, and when they want to retrieve a document, the document is sent from the object server that is local to them (in network topology terms).

We should consider the possible disadvantages with this architecture. Aside from the hardware administration overhead, there are few disadvantages to having multiple object servers within the same data center; however, there are some issues to consider when distributing object servers across geographies.

- ▶ The access to documents by the users must be carefully considered before deciding on geographically distributed object servers. If a significant number of users require documents from object servers that are not local to them (in networking terms), there are negative performance effects.
- ▶ Wide area networks (WANs) are often less reliable than local area networks (LANs). If object servers are inaccessible from the library server, then, although documents might be physically located in the same building as the users, the users will not be able to access the documents.

6.2.2 OnDemand configuration

The way in which reports are defined, indexed, and stored within OnDemand greatly influences the speed at which OnDemand can retrieve them. There are variety of hints and tips for the optimum way of defining reports within OnDemand that are described in Chapter 2, “Administration” on page 21. From the perspective of performance, areas of specific interest in that chapter are “Large Object support” on page 31 and “Field Information” on page 27.

OnDemand system logging

An area that is not discussed in Chapter 2, “Administration” on page 21, is the effect of system logging on the overall performance of OnDemand.

Four system logging event points can be selected to control the messages that OnDemand saves in the system log. OnDemand can record a message when the following events occur:

- ▶ **Login:** A user logs on a server.
- ▶ **Logoff:** A user logs off a server.
- ▶ **Application Group Messages:** A user queries or retrieves application group data and other types of application group events.
- ▶ **Failed Login:** An unsuccessful logon attempt is made.

The amount of logging that is done on a server is controlled in two places: in the *System Parameters* and in the *Message Logging* tab of each of the application groups on the system. By default, when you create a new application group, all of the message logging is turned on (except for the database queries). Also, within the system parameters, all four of the system logging event points listed earlier are checked by default on a standard OnDemand installation. This means, by default, an OnDemand server is in *full* logging mode. In some cases, this can have significant effects on the system performance.

Unless you are running the system in a diagnostic mode, we recommend that you turn off logging for *Login* and *Logoff* messages. If it is an active system with large numbers of active users constantly logging in and off, this should improve the performance of *Login* times.

6.2.3 Database

OnDemand creates a database as part of the installation process. Apart from the ability to choose the location of the database logs, there is little opportunity to change the default values used by the database manager. For example, in the case of DB2 Universal Database, the default parameter values are oriented toward small machines with small amounts of memory. It is common to alter some of the default parameters to optimize performance in a specialized environment.

Database tuning efforts should be carefully monitored by a qualified database administrator and are highly dependent on individual system resources. The parameters presented in the following sections should serve as examples of areas within a database configuration that can be tuned if database performance problems are identified.

In the following sections, we provide guidance and recommendations. We have strictly avoided specifying actual values for these parameters because each tuning effort is different depending on the environment on which the database resides.

Memory

It is possible to allocate system memory to a database application in order to gain more control over the way in which system resources are allocated. In DB2, this memory allocation is called a *buffer pool*, and each database has at least one of these. All buffer pool resides in global memory, which is available to all applications using the database. If the buffer pools are large enough to keep the required data in memory, less disk activity will occur. Conversely, if the buffer pools are not large enough, the overall performance of the database can be severely curtailed and the database manager can become I/O-bound.

In DB2, the default buffer pool size is 1000 (specified in 4 KB pages), which is 4 MB. We recommend that you increase this value enough to reduce the risk of becoming I/O-bound and give the database a decent buffer pool to work with. Depending on the system resources, it is beneficial for a production environment to increase the buffer pool to limit I/O contention.

Query optimization

Query optimization determines how much effort the database puts into optimizing queries. A high value is often used in a data warehousing environment. Lower values are useful in an online transaction processing (OLTP) type environment that uses simple dynamic queries, which is the case with most of the OnDemand systems. Lowering this parameter can significantly reduce CPU activity and prepare time. In DB2, for example, the default query optimization class (DFT_QUERYOPT) is set to a value of 5, which is regarded as significant query optimization.

Open database files

Open database files determines the maximum number of file handles that can be used for each database agent. For more information about setting the number of database agents for OnDemand, see 6.2.2, “OnDemand configuration” on page 163.

In DB2, the parameter for setting the maximum number of open database files per application is called MAXFILEOP and the default value is 64. If opening a file causes this parameter to be exceeded, some files in use by this agent must be closed. Increasing this parameter helps to alleviate the overhead of excessive opening and closing files. The default tablespace type for an OnDemand installation is SMS. Generally, SMS tablespaces need a larger value. SMS tablespaces have at least one file per database table (usually more).

Locking

Locking is the mechanism that the database manager uses to control concurrent access to data in the database by multiple applications. The database manager imposes locks to prohibit applications from accessing uncommitted data and thus protects data integrity. In DB2, for example, the parameter controlling this is LOCKLIST, and the default is 100 (specified in 4 KB pages). Setting the lock list parameter sufficiently high helps to prevent lock escalation. *Lock escalation* is the conversion of record locks to table locks.

Transaction logs

The database transaction logs should be placed on a separate physical disk or disks from the database to avoid contention with the data I/O. In addition, physically separating the database transaction logs from the database means that in the event of a media failure, the logs are not lost with the database and recovery is possible. The log disks must be protected to ensure database consistency, and due to the high write content, mirroring is typically preferred over RAID-5. Notice that there is a performance hit to mirroring.

DB2 parallelism

Parallelism within DB2 is designed for best performance when running few, but complex, number-crunching type queries. OnDemand submits a very large number of simple queries for user logon and folder searches. The overhead generated by DB2 parallelism can impact server performance in an OnDemand environment.

To check the setting for parallelism, do the following:

1. From a DB2 command prompt, enter the following command:
`GET DATABASE MANAGER CONFIGURATION`
2. Check the INTRA_PARALLEL = parameter. If this parameter is set to YES, change it to No by enter the following command:
`UPDATE DATABASE MANAGER CONFIGURATION USING INTRA_PARALLEL NO`
3. Restart DB2 for the change to take effect.

6.2.4 System management

The recommendations and guidance in this section fall under the category of system management because the majority of them are outside of the control of the OnDemand product code.

Running parallel `arsload` jobs for PDFs

A common misconception with OnDemand is that because there is a single load program for loading data into OnDemand (`arsload`), you might only have one of these load jobs running at any one time. This is *not* true. Depending on the resources of the machine (memory, CPU, and hard disk drive), it is possible to run multiple `arsload` jobs simultaneously to improve the overall performance of the load process. This is especially true in an architecture where there are multiple object servers distributed onto different physical machines.

Our key recommendation with regard to running multiple `arsload` jobs is that, unless the system has a distributed object server architecture, try not to start all of the load jobs at the same time. If you start multiple `arsload` jobs simultaneously, you might still get a performance benefit. If you stagger the load jobs, then each of them will be at a different phase of the load process, which will maximize performance due to less I/O contention in the database and in cache storage volumes. To clarify this point further, see Figure 6-1 on page 160, which illustrates the different phases in the load process.

The ARS_NUM_DBSRVR parameter

The ARS_NUM_DBSRVR parameter is set in the `ars.cfg` file. It determines the number of DB2 database agents that are launched automatically when the OnDemand server is started. Under normal circumstances, each connection to the database requires a database agent, which is launched whenever a database connection is requested. However, starting several database agents that remain active and waiting for connection requests can enhance performance because there is no time spent initializing and then closing the agents.

For systems that are running several large load jobs in parallel, or for systems that have large numbers of active users, we recommend that you increase this parameter from the default of 4. For more detailed guidance, refer to Appendix A, in *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232.

File systems on UNIX

During the installation and setup of OnDemand, one of the tasks is to create the file systems that are required to contain the various OnDemand components. This task is described in “Disk storage devices on a UNIX server”, in Chapter 3, “Disk storage requirements”, in the manual, *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232. Within this guide, Table 14, “Disk storage group for a large organization”, gives an example of the file systems which you might be required to create for a large OnDemand system. More importantly, it gives an example of how to organize these file systems to optimize performance.

The reason this is so important is that I/O contention is one of *the most* common causes of performance problems.

For performance reasons, when the OnDemand file systems are created, the following components *should not* be on the same physical media:

- ▶ The cache file system
- ▶ The database file system
- ▶ The primary logs file system
- ▶ The secondary logs file system
- ▶ The load / indexing file system
- ▶ The OnDemand temporary space file system

Operating system performance

With the wide variety of platforms that OnDemand supports, there comes an even wider variety of performance issues specific to the individual architecture of these underlying operating systems. Depending on the operating system itself, there are a number of different tools and methods for monitoring and tuning the operating system.

If you are experiencing performance problems and you believe that you have followed all of the guidance available for configuring OnDemand, make sure that you check with the vendor of your operating system to confirm that you are at the latest service pack or maintenance level. Also ensure that there are no known problems with performance at this level.

Note: A virtual memory leak, discovered in AIX 4.3.3, is described in APAR IY15250. The problem is fixed by applying maintenance level 9, and by tuning the MINFREE and MAXFREE parameters. This AIX performance problem is a good example regarding the necessity to check operating system maintenance for performance fixes.

Network

OnDemand has various features, such as compression and large object support, which minimize the impact of retrieving large quantities of information from the server over a network. However, network performance and topology can often be the bottleneck in a system architecture, especially when the data retrieved is large image files that cannot be compressed. For guidance in tuning the OnDemand environment to cater for the network bandwidth that is in place, see the following sections:

- ▶ 6.2.1, “OnDemand architecture” on page 162
- ▶ 6.2.2, “OnDemand configuration” on page 163
- ▶ 6.3, “Performance issues based on data type” on page 177

6.2.5 Storage management

Regardless of the platforms, storage management with OnDemand can be divided into two areas: cache storage managed by OnDemand and archive media managed by an external product such as Tivoli Storage Manager, object access method (OAM), Virtual Storage Access Method (VSAM) or Archive Storage Manager. In terms of storage management, one of the key performance features with OnDemand is the ability to load data to archive media, but simultaneously retain a temporary cached copy of the most recent archived data on fast access media (such as the hard disk drive). The expiration and management of this cached copy of the data is done by OnDemand. After a certain predefined period has elapsed, the data is removed from cache and the only remaining copy will be held on the much slower archive media managed by either Tivoli Storage Manager, OAM, VSAM or Archive Storage Manager 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 the way in which the archive media manager is configured. To ensure that the OnDemand configuration done by the administrator is *not* causing the slower performance from the storage manager, see 6.2.2, “OnDemand configuration” on page 163.

Physical media and library issues

To troubleshoot possible performance problems with physical media and library devices, it is important to understand the basic principles behind the working of these devices. Typically, a library device that contains either optical or tape volumes has a certain number of drives. A *drive* is an area in the library where a volume is placed in order to be read from. For example, if an IBM 3995 Optical jukebox has six drives and all of these drives are busy reading from volumes, then the next time a request is made for data to be read from another volume, that request is queued.

This example is typical of a situation where data is loaded into OnDemand at the same time that users are active on the system and retrieving documents from OnDemand. The most common way to avoid this performance problem is to schedule load jobs at times when the system is not in use by the user community. This way the load can have full use of all of the drives in the library to load the large quantities of data that are necessary during this process.

Mount retention of optical and tape volumes

Mount retention determines how long in minutes to retain idle storage volumes in a drive before dismounting. This parameter can improve response time by leaving previously mounted volumes online. A low number is recommended for high-access systems. However, setting the mount retention value depends on the way in which archived data is searched. For example, if a number of users are all retrieving data from a similar time period, then it is likely that this data will be on the same physical volume. Leaving the volume mounted in the drive between retrievals will improve retrieval times, because time is not spent finding and mounting the volume onto the drive before the data can be read.

In Tivoli Storage Manager, the mount retention of optical or tape volumes is set when defining the device class. Example 6-1 is an extract from the archive.mac file, which is shipped with a standard installation of OnDemand for Multiplatforms. The archive.mac file is an extremely useful tool to use in order to configure a Tivoli Storage Manager environment with OnDemand. For more information about storage management with Tivoli Storage Manager, OAM (for zSeries) and Archive Storage Manager (for iSeries), see Chapter 5, “Storage management” on page 105.

Example 6-1 Setting mount retention in Tivoli Storage Manager

```
def devclass odlib0 devtype=optical format=1300MB library=archlib0
mountlimit=2 mountretention=10 estcapacity=1309M
```

6.2.6 ODWEK configuration

The OnDemand Web Enablement Kit has a number of functional components that can be configured within your environment to optimize a thin client search and retrieval from an OnDemand server. For more information about the ODWEK architecture, application programming interfaces (APIs), and installation, see Chapter 8, “OnDemand Web Enablement Kit” on page 199.

The ODWEK cache

One of the performance enhancing features of ODWEK is the ability to cache session information and even documents on the Web server machine. However, to ensure that you are optimizing ODWEK performance while minimizing the amount of hard disk drive required for the ODWEK cache, it is important that it is configured and sized correctly.

To know how to size the cache, it is important to understand the type of information and data files are stored. Table 6-2 on page 171 describes the different types of data that can be stored in the cache as well as explaining how many of these different data files are created for each user.

Table 6-2 Cached data description

Description	Frequency	File extension
Server name	One for each user	SRV
Folder names	One for each user	FNM
Folder search fields	One for each folder for each user	FLD
AFP resources	One for each resource	RES
Documents	One for each document	DOC
OnDemand internal file	One for each document	IDOC
Line data document	One for each line data doc	ASC

It is impossible to estimate a generic size or growth rate for all ODWEK systems because the numbers of users, data types, folders and Advanced Function Presentation (AFP) resources vary enormously from one installation to another. However, it is possible to offer guidance, given information about the individual environment.

ODWEK cache sizing process

Despite the variability of data produced in the cache, it is possible to apply the following simple process to size the ODWEK cache:

1. Install and test ODWEK based on the instructions in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231.
2. In the arswww.ini file, ensure that the Cache* parameters (Example 6-2) are set.

Example 6-2 Cache parameters

```
[configuration]
Language=ENU
TemplateDir=/home/httpd/docs/templates
ImageDir=/images
AppletDir=/applets
TempDir=/odwek/temp
CacheDir=/odwek/cache
CacheSize=10
CacheMinThreshold=40
CacheMaxThreshold=80
CacheDocs=0
CacheUserIDs=web,demo
```

3. After testing the installation, there should be a number of files in the location described by the CacheDir parameter. Delete all of these files so that the cache is empty.
4. Based on knowledge of the use of OnDemand by the user community, logon to OnDemand via ODWEK and perform a number of search and retrieve operations from a typical set of folders and different data types.
5. Examine the files that reside in the cache directory. You should see several files. The reason for the existence of the files shown is described in Table 6-2 on page 171.

For example in Figure 6-3, you can see that the **admin** user logged on to OnDemand via ODWEK and opened three folders (Credit Card Statements, Letters, and TWS Job Reports).

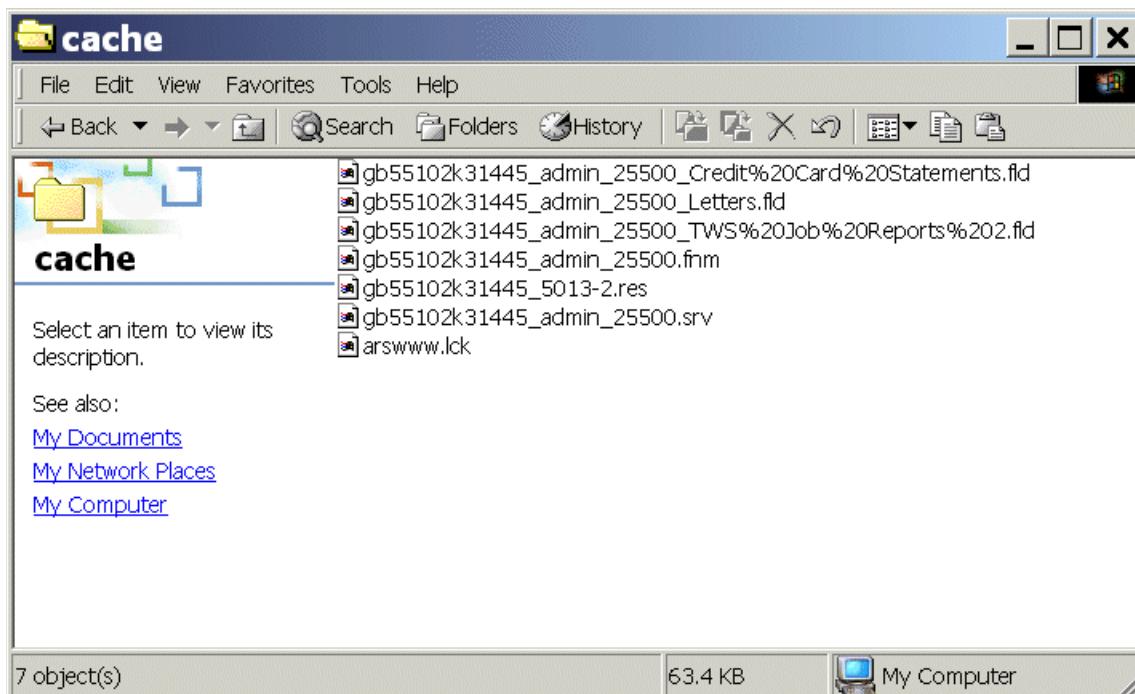


Figure 6-3 The ODWEK cache

6. Figure 6-3 also shows an example of the overall size of the cache after doing this operation (63.4 KB). If you are satisfied that this is a typical search, then multiply the collective size of these files by the number of concurrent users expected to access this ODWEK system.
7. We recommend that you also add 25% to the value from step 6 for contingency.

Attention: Notice from Example 6-2 that the CacheDocs parameter was set to 0; therefore, only folder, server, and resource information was collected in the cache illustrated in Figure 6-3.

Caching documents

If you intend to cache documents that are retrieved from OnDemand in conjunction with the session information and resources, then it is far more difficult to estimate an optimum cache size. You can follow the same process as in the previous section; however, be aware of the following factors:

- ▶ Documents are typically much larger than session information, and so when sizing a cache to include documents, you will find that much more space is required.
- ▶ If the likelihood of many users frequently viewing the same documents is high, then caching documents can be beneficial because ODWEK is not required to retrieve the documents from OnDemand many times.
- ▶ If the cache size is too large, then it might take ODWEK more time to check the cache to see if a document is present than to retrieve the document from OnDemand. To avoid this, we recommend that you set the cache size no larger than 200 MB.
- ▶ If the network is slow or overloaded between the OnDemand server and the Web server, then caching documents might alleviate this problem.

The ODWEK cache is one of the main areas for tuning ODWEK to optimize performance in your environment. Considered and deliberate sizing of the cache can see significant performance benefits.

Servlet verses CGI

As described on “Deploying the CGI program” and “Deploying the servlet”, in Chapter 2, “Installation and Configuration”, in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231, you can optionally choose one of two protocols to control ODWEK:

- ▶ **CGI program:** The CGI program runs against an HTTP server, such as the IBM HTTP Server.
- ▶ **Java servlet:** The servlet runs on a Java-enabled HTTP server with a Java application server, such as the IBM WebSphere Application Server.

From a performance perspective, we recommend the servlet implementation of ODWEK. The reason for this is that the CGI program is called by the Web server program each time an operation is submitted by a user, which involves connecting to the OnDemand server. When the CGI program is executed, it must load the shared libraries into memory, and this memory must be allocated to

each of the CGI processes that have been launched. In contrast, the servlet must only load the shared libraries into memory the first time it is activated per session. Subsequent users that submit operations to OnDemand via the servlet should experience faster response times than the CGI.

Web server architecture

A common method to perform workload balance on a system is to distribute components of the architecture across multiple physical machines. A good example of this is demonstrated by the OnDemand architecture, which is described in 6.2.1, “OnDemand architecture” on page 162.

When ODWEK is used as the viewing technology for OnDemand, the Web server must be considered as one of those components within the architecture to separate from the machine on which the OnDemand server is running. If the Web server and the OnDemand server are separated onto dedicated machines, then this should significantly improve overall system performance.

It is also possible to have multiple Web servers. This is necessary if you have an OnDemand architecture with multiple distributed object servers, where typically there is a Web server (with ODWEK installed on it) in close proximity to each of the object servers. With this configuration, ODWEK users can benefit from the fast retrieval of their documents from object server in their location.

Tip: With ODWEK, the OnDemand object server must deliver the object *via* the Web server. If you have multiple object servers that might be distributed throughout your enterprise but only a single Web server, then all objects must be delivered through that Web server. *You are not using the strengths of the fast document retrieval from the distributed object servers.*

Debugging and logging turned off

For the purposes of auditing operations and debugging any problems discovered with ODWEK, a debug and logging tool can be activated. The parameters in Example 6-3 produce a file called arswww.log in the specified directory. This is an excellent tool for debug purposes, but for performance reasons, this logging *must* be turned off in a production environment. This is done by setting the log parameter to 0.

Example 6-3 Logging in ODWEK

```
[DEBUG]
log=1
logdir=D:\temp
```

User IDs

For OnDemand installations intended for use only by internal employees, the most common use of the OnDemand user ID is for each individual who requires access to OnDemand to have a unique ID. Typically within an internal environment, there is an easily defined user who can be categorized into OnDemand groups based on department or job responsibility. Under these circumstances, ODWEK is simply used as a thin client access technology to the OnDemand server and is for internal use only.

However, the majority of ODWEK installations are intended as a method of externalizing access to documents stored in OnDemand to a public Internet site. Depending on the nature of the content that you intend to deliver to a Web site, it might be far more difficult to define each individual user who may attempt to access the information. Also, more importantly from a performance and ease of administration perspective, you may not need or want to define each person who requires access to information of an individual OnDemand account.

Figure 6-4 describes the process that is followed within ODWEK when a user with their own OnDemand user ID retrieves a document. In reality, this process is followed each time the user issues a new logon; for subsequent searches after a logon, there is ODWEK caching for *folder lists* and *folders* as described in “The ODWEK cache” on page 170. Figure 6-4 shows a simple example of an interaction between a user and OnDemand, which involves four requests submitted by the user via ODWEK and four replies submitted by OnDemand via ODWEK.

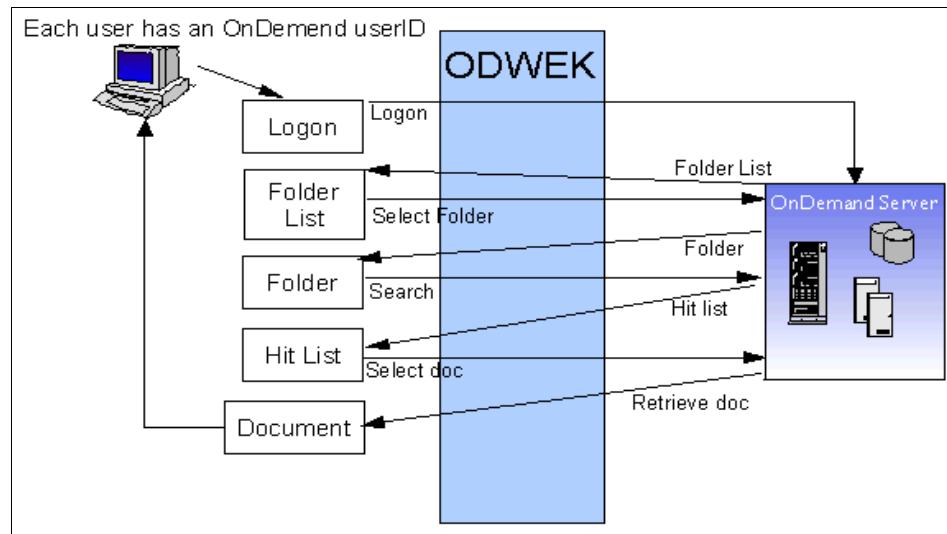


Figure 6-4 ODWEK with multiple user ID access

We recommend that you optimize the configuration shown in Figure 6-4 on page 175 by defining a small subset of users within OnDemand and forcing all Web users to access OnDemand via this subset of OnDemand user. The main advantage of channeling large numbers of Web requests through a small number of users is that ODWEK is far more efficient at managing a small number of highly active users than a very large number of idle users.

Figure 6-5 shows the significant decrease in communication between ODWEK and the OnDemand server when a single user ID is used as a channel for multiple user requests. The configuration illustrated in Figure 6-5 is typically implemented with the use of a custom Web application that the public Web user uses to obtain information stored in OnDemand. This Web application authenticates the identity of the Web user and controls the access that they have to OnDemand. Rather than using unique user IDs, the Web application accesses OnDemand via a common user ID (or a small subset of user IDs) and retrieves the required information on the Web users' behalf by constructing queries dynamically and submitting them to ODWEK for processing. By using this technique, not only is performance of OnDemand and ODWEK optimized, but an added layer of Web security is put in place to protect access to the OnDemand server.

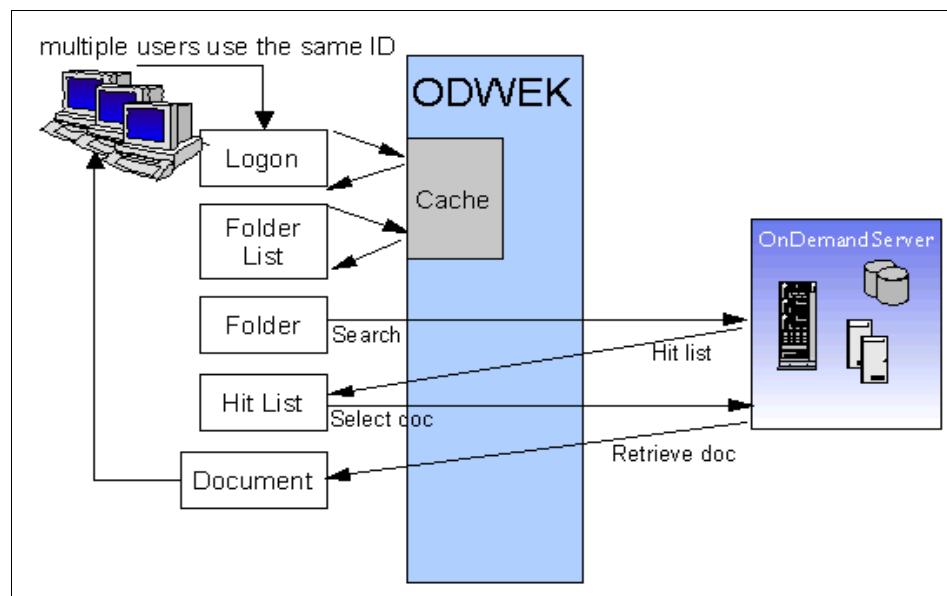


Figure 6-5 ODWEK with single user ID access

If ODWEK is used to deliver content to a public Web site, the configuration illustrated in Figure 6-5 on page 176 is recommended for the following reasons:

- ▶ There is a substantial decrease in the workload of OnDemand users or system administrators due to a significantly smaller user community.
- ▶ Unnecessary work by the Web users to create OnDemand user accounts can be avoided.
- ▶ There is a substantial decrease in the size of the ODWEK cache.
- ▶ Communication between ODWEK and the OnDemand server can be decreased by a factor of two, therefore improving response times experience by the users.

6.3 Performance issues based on data type

This section describes issues that are related to individual data types that can have significant effects on the overall performance of OnDemand. Some of these issues can be addressed by selecting (or deselecting) certain functions and features within OnDemand. Some of the issues that we discuss can only be addressed by changing the way in which the data is produced from the source.

6.3.1 PDF data

Portable Document Format data is an increasingly common data type that can be archived within OnDemand. The key advantages of using this data type as a document format are as follows:

- ▶ 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 is free to download from the Adobe Web site and the browser plug-ins for PDF are also freely available.

From a performance standpoint, the issue that commonly arises with the PDF data type is illustrated in Figure 6-6 on page 178. As we have stated previously, one of the main advantages of PDF data is its self contained nature; however, when archiving this data, it is also one of the main disadvantages.

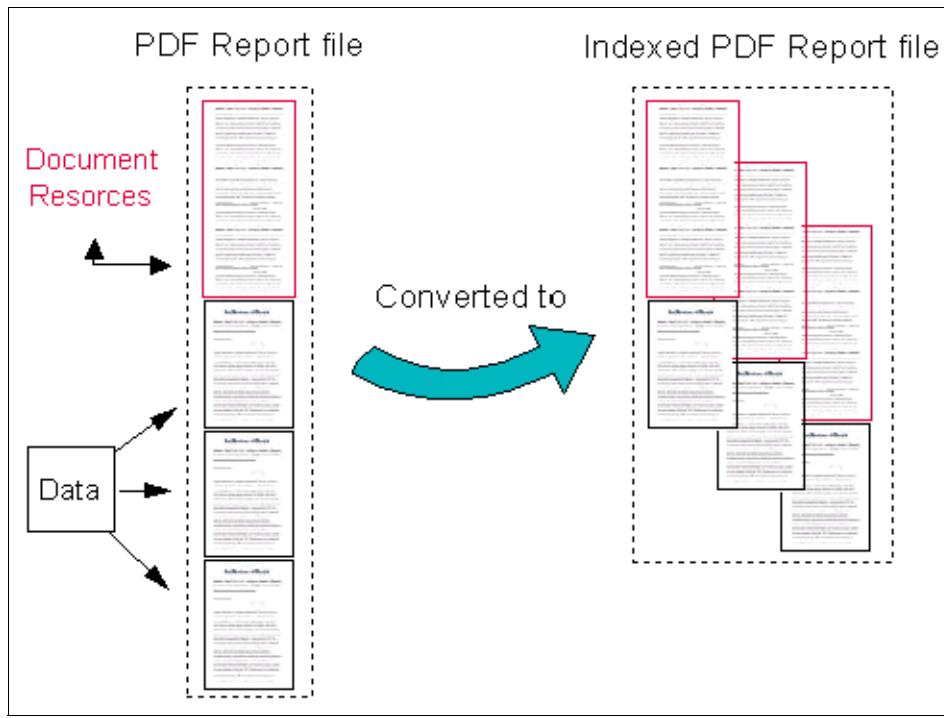


Figure 6-6 PDF architecture

When a PDF is produced, such resources as images and custom fonts are placed in the data stream once (usually at the top) and then referenced many times from within the PDF file. If for example, a large report is produced that might be a collection of many small documents, then the advantage is that only one copy of the resources are required. However, in order for OnDemand (or any archive product) to split this report into a collection of self contained documents, the resources must be copied and placed within each of the smaller documents. The effect of doing this is illustrated in Figure 6-6. It is common for the sum of the individual documents to be many times larger than the original report.

This issue can create a variety of problems during the load process:

- ▶ If this increase in file size has not been anticipated, the temporary space used during indexing can be too small and the load will fail.
- ▶ The PDF indexer has a maximum file size, 4 GB, that it can load to OnDemand. If the resulting PDF file that the indexer produces is larger than this maximum file size, then the load fails.

The most common cause for exponential increases in the PDF file to be loaded into OnDemand is the inclusion of custom fonts into the PDF data. For a small

two to three page document, custom fonts typically make up the majority of the overall size of a document if they are included in the data stream.

The primary goal of this section is to increase awareness of the issues with archiving PDF data. Our recommendation is that if PDF data is the only format possible to archive your data, then wherever possible, use the base 14 fonts, which do not need to be included in the data. For more information about the PDF data stream and the font issues, see Chapter 7, “PDF indexing” on page 185.

6.3.2 Line data

Line data (ASCII or EBCDIC text-based reports) is the most common type of data stored in OnDemand. The type of line data that we discuss 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 typically has a transaction number that appears on every line and must be sorted either by column or row and either ascending or descending.

When indexing transaction data, if each transaction number from each line of the report is treated as a database index, such as date or customer name, then the database grows to be extremely large in a short period of time. OnDemand has a special type of field for transaction data, which is illustrated in Figure 6-7 on page 180 by the boxed data on the left of the window.

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. OnDemand queries the database by comparing the search value entered by the user to two database fields, the beginning value and the ending value. If the value entered by the user falls within the range of both database fields, OnDemand adds the item to the document list.

From a performance perspective, using the transaction data field for transaction style line data optimizes indexing performance by significantly reducing the number of index values to be inserted into the database. This means that loading and retrieving these extremely large reports is significantly faster and the OnDemand database is many times smaller.

BANK A		UNMATCHED ITEMS BY SEQ						
CENTRAL ADJUSTMENTS DEPARTMENT								
INCOMING SEQ NO.	OUTGOING ID	KB SEQ	ENDPOINT ID	DT	ITEM AMOUNT	CASH LETTER AMOUNT	ROUTING TRANSIT	A
P000000072	21593.34	1	0000-0032	TR	50.00	21593.34	0000-0371	7983
P000000073	2151.39	1	0000-0194	TR	50.00	2151.39	0000-0040	450283
P000000074	2151.39	2	0000-0194	TR	20.00	2151.39	0000-0040	450294
P000000075	2151.39	3	0000-0194	TR	10.00	2151.39	0000-0040	450281
P000000076	2151.39	4	0000-0194	TR	40.00	2151.39	0000-0040	450281
P000000077	2151.39	5	0000-0194	TR	296.00	2151.39	0000-0040	450281
P000000078	2151.39	6	0000-0194	TR	77.33	2151.39	0000-0040	450281
P000000080	2151.39	7	0000-0194	TR	127.00	2151.39	0000-0040	450281
P000000081	2151.39	8	0000-0194	TR	25.00	2151.39	0000-0040	563198
P000000082	2151.39	9	0000-0194	TR	135.00	2151.39	0000-0040	450280
P000000084	2151.39	10	0000-0194	TR	300.00	2151.39	0000-0040	212909
P000000085	2151.39	11	0000-0194	TR	25.00	2151.39	0000-0040	212909
P000000086	2151.39	12	0000-0194	TR	11.00	2151.39	0000-0040	450281
P000000089	2151.39	13	0000-0194	TR	206.00	2151.39	0000-0040	450280
P000000091	8175.12	1	0000-0372	TR	264.75	8175.12	0000-7083	0
P000000093	2151.39	14	0000-0194	TR	233.00	2151.39	0000-0040	45284
P000000094	2151.39	15	0000-0194	TR	96.90	2151.39	0000-0040	563206
P000000095	1802.24	1	0000-0502	TR	638.00	1802.24	0000-1544	9
P000000097	21593.34	2	0000-0032	TR	341.54	21593.34	0000-0589	5530
P000000165	1802.24	2	0000-0502	TR	1164.24	1802.24	0000-1512	1669

Define a Transaction Field [Display] Page 1 of 100 100%

Figure 6-7 Transaction data in the graphical indexer

6.3.3 AFP data

Advanced Function Presentation (AFP) data is a multi-part data type. This means that in addition to the variable data itself, there are also external resources, such as images, fonts, and logos, which are referenced by the AFP data stream. When OnDemand stores AFP, 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. The fact is that 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 ATM or TT fonts. OnDemand provides this mapping for most standard fonts. For more information about mapping custom fonts, refer to *IBM Content Manager OnDemand - Windows Client Customization Guide and Reference*, SC27-0837.

One possibly useful implementation of storing fonts with the resource group is where server reprint is necessary. If the fonts are stored with the resource group, they can be retrieved from 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 only serves to increase network traffic when transferring the resource to the PC. 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. As long as 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). Using this approach also allows all AFP data that references the font to use the single instance of the font without redundant inline storage.

Figure 6-8 shows the panel in the application where you can select the resources to collect. Unless reprints to AFP printers with 100% fidelity is a requirement, we recommend that fonts are not collected.

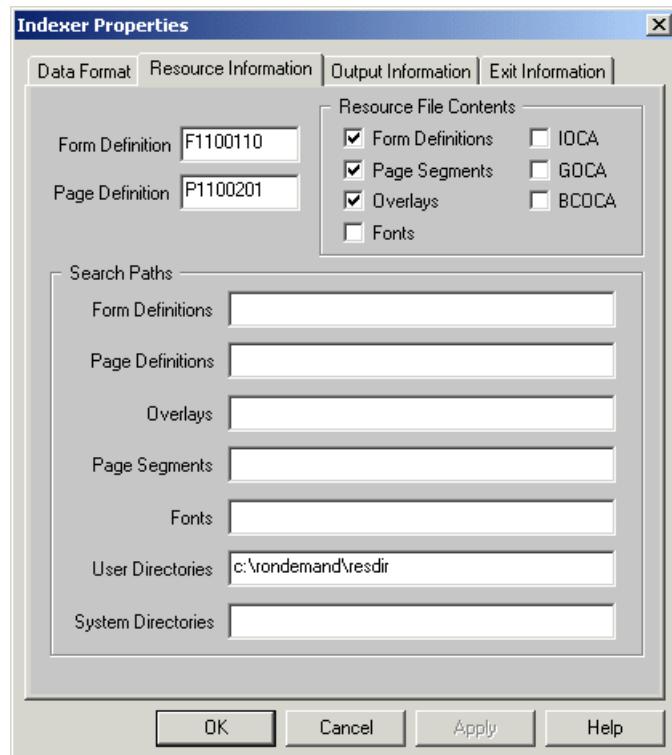


Figure 6-8 Collecting AFP fonts

The iSeries server does not collect the fonts, nor does it give the administrator that option. The Resource Information window (under Indexer Properties) is not available to the iSeries administrator. If you are reprinting to an AFP printer, the fonts must be available on the iSeries server, or font substitution is done.

Image data

To optimize performance with storing and retrieving image formats such as TIFF, GIF, and JPEG, we recommend that you do not compress the data because the file sizes might actually increase. To turn off compression, select the Disable option from the Load Information tab within the application (see Figure 6-9).

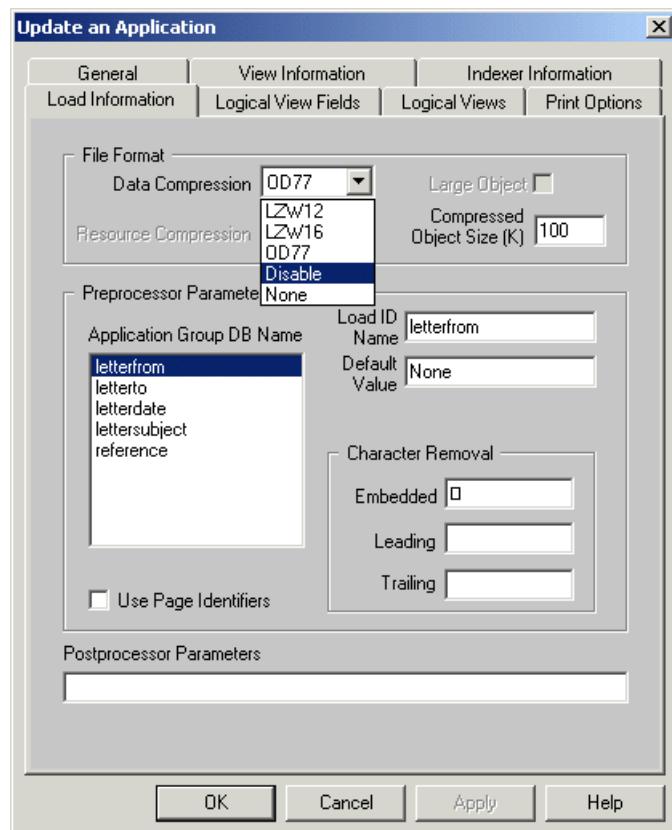


Figure 6-9 Disabling compression

Notice that there are two options that turn off data compression:

- ▶ **Disable:** OnDemand does not compress the input data. Choose this option when the input data, such as PDF and compressed TIFF, is already compressed. Documents are uncompressed by the appropriate viewer on the client (for example, Acrobat Reader).
- ▶ **None:** OnDemand does not compress the input data when loading it into the system. When the user selects a document for viewing, OnDemand compresses the document before transmitting it over the network and uncompresses the document at the client.



PDF indexing

In this chapter, we describe Portable Document Format (PDF) indexing from a practical standpoint, with specific reference to the issues surrounding the archival of the PDF data stream. We discuss the process of graphically indexing PDF and any differences between the respective platforms in this area.

In this chapter, we cover the following topics:

- ▶ Getting started
- ▶ Indexing issues with PDF
- ▶ PDF indexer
- ▶ PDF indexing on z/OS

7.1 Getting started

PDF is a standard specified by Adobe Systems, Incorporated, for the electronic distribution of documents. PDF files are compact; can be distributed globally via e-mail, the Web, intranets, or CD-ROM; and can be viewed with the Acrobat Reader.

The following sections serve as an introduction to the technical concepts and architecture behind the PDF data type to help you understand the potential issues in indexing this data.

7.1.1 What is the Portable Document Format?

In simple terms, PDF is a data type or file format that is independent of the platform on which it is created. A PDF file contains a PDF document and the resources reference by that document.

Type 1 fonts

Type 1 fonts, described in detail in *Adobe Type 1 Font Format*, by Adobe Systems Incorporated, are special-purpose PostScript® language programs used for defining fonts. They use a specialized subset of the PostScript language for more compact representation and optimized performance.

Compared with the larger Type 3 fonts, Type 1 fonts can be defined more compactly due to the fact that they make use of a special procedure for drawing the characters that results in higher quality output at small sizes and low resolution. They also have a built-in mechanism for specifying hints, which is data that indicates basic features of the character shapes not directly expressible by the basic PostScript language operators.

The base 14 Type 1 fonts

For every PDF data stream, there exists a core set of fonts that are guaranteed to be available to the Acrobat program. These fonts are the PostScript names of 14 Type 1 fonts, known as the “base 14 fonts”:

- ▶ Courier
- ▶ Courier-Bold
- ▶ Courier-BoldOblique
- ▶ Courier-Oblique
- ▶ Helvetica
- ▶ Helvetica-Bold
- ▶ Helvetica-BoldOblique
- ▶ Helvetica-Oblique
- ▶ Times-Roman

- ▶ Times-Bold
- ▶ Times-Italic
- ▶ Times-BoldItalic
- ▶ Symbol
- ▶ ZapfDingbats

Because the only external resources for a PDF document are the base 14 fonts, each PDF document embeds any other fonts that are not members of the base 14 fonts. In the event that a PDF file has a company logo or image on a set of pages, that logo or image is also embedded within the document. Barcode fonts are embedded within the PDF document as well.

7.1.2 PDF and the OnDemand client

If you plan to use the Report Wizard or the graphical indexer to process PDF input files, then you *must* first install Adobe Acrobat on the PCs from which you plan to run the administrative client. You must purchase Adobe Acrobat from Adobe.

Notes:

- ▶ Adobe Acrobat Approval, an alternative to Adobe Acrobat, was previously offered by Adobe. It is no longer sold or supported by Adobe as of September 2004. For information about transitioning to Acrobat Reader Extensions Server or Adobe Acrobat, refer to the following Web address:
<http://www.adobe.com/products/acrapproval>
- ▶ Users access PDF documents through OnDemand clients depending on the OnDemand parameters and software installed on the workstations.

OnDemand provides the ARSPDF32.API file to enable PDF viewing from the client. If you install the client after you install Adobe Acrobat, then 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, then you must copy the API file to the Acrobat plug-in directory manually. Also, if you upgrade to a new version of Acrobat, then you must copy the API file to the new Acrobat plug-in directory.

The default location of the *API file* is \Program Files\IBM\OnDemand32\PDF. The default *Acrobat plug-in directory* is \Program Files\Adobe\Acrobat x.y\Acrobat\Plug_ins. Here, x.y is the version of Acrobat, for example, 4.0, 5.0, and so forth.

7.2 Indexing issues with PDF

When indexing PDF data, you might experience surprising results related to the size of the files created by the OnDemand indexer after it indexes the data. In some cases, the PDF file that is loaded into OnDemand is many times larger than the source PDF file.

Here is an example of why the initial PDF file size can grow to be multiple times larger than the original. Suppose that OnDemand receives a PDF data stream with the following characteristics:

- ▶ A 100-page PDF file includes one non-base 14 font.
- ▶ A company logo is displayed on every fifth page.
- ▶ A barcode font is displayed on every fifth page that describes the customer account number.
- ▶ The file is 100 KB in size, and OnDemand is required to index the document into twenty 5-page documents that represent twenty customer accounts with five pages of customer information each.

The output PDF file is generated from the original PDF in the following manner:

- ▶ Each of the 20 PDF documents include any and all fonts that are not members of the base 14 fonts. OnDemand has 20 copies of any non-base 14 font in the resultant indexed PDF file. The 10 KB of font becomes 200 KB worth of fonts.
- ▶ Each of the 20 PDF documents has a copy of the company logo instead of one copy of the company logo for the non-indexed 100-page document. A compressed image of 25 KB is now 500 KB.
- ▶ Each of the 20 PDF documents has a copy of the barcode font required to print the customer account information instead of one copy for the non-indexed 100-page document. A barcode font of 5 KB is now 100 KB.

As you can see, the original 100 pages with one company logo, one barcode font, and one non-base 14 font has become 20 wholly contained PDF documents with one copy of the non-base 14 font each, one copy of the barcode font each, and one copy of the company logo each.

The indexed file size is around 860 KB, 60 KB being the actual text in the 100 KB original file, and 40 KB being the resources in the original file, which expanded to 800 KB through duplication. Considering that the logo, the barcode font, and possibly the non-base 14 font are images, you can see how the file size multiplies. Due to the extraction of the fonts, logos, barcode, and pages into separate documents, the result is that extra hardware, disk, and RAM are required to accomplish this task. For a discussion of what this means in terms of

performance, as well as an illustration of this process, see 6.3.1, “PDF data” on page 177.

7.2.1 Listing fonts in a PDF file

As previously discussed, the main cause of the PDF file size increasing after indexing is due to the embedded fonts that are not part of the base 14 Type 1 font families. If you are experiencing problems with PDF file sizes growing and you suspect that it is the number of custom fonts in your PDF data, then there is a simple way within the Adobe viewer to list the fonts in your data.

To list the fonts in a PDF:

1. Display your PDF document in the Adobe viewer.
2. Click **File → Document Properties → Fonts...** You should see a list of fonts for the document such as the one shown in Figure 7-1.
3. If the PDF file you used is large, the fonts that are displayed are not a complete list for the entire report. Click **List All Fonts...** to obtain a full list.

Document Fonts			
Fonts in: 4011.pdf			
Original Font	Type	Encoding	Actual Font
Helvetica	Type 1	Windows	ArialMT
C39Hrp24DmTt0199.6259227	Type 1	Custom	Embedded Subset
Helvetica-Bold	Type 1	Windows	Arial-BoldMT
Times-Roman	Type 1	Windows	TimesNewRomanPSMT
Times-Bold	Type 1	Windows	TimesNewRomanPS-BoldMT
Bookman-DemiItalic	Type 1	Windows	Adobe Serif MM
C39P12DmTt099.8139322	Type 1	Custom	Embedded Subset

List All Fonts...

Figure 7-1 Listing fonts in a PDF document

Tip: When indexing, OnDemand uses the Adobe parsing technology to produce the output PDF files. If you are experiencing poor performance during indexing, you can do a stand-alone Adobe parsing test by clicking **List All Fonts...** as shown in Figure 7-1. If this operation is fast, then it is an OnDemand problem; if it is slow, then the performance issue is caused by the PDF data stream.

7.3 PDF indexer

The PDF indexer can process an input file that contains up to 2.1 billion pages, so long as the input file does not exceed 4 GB in size. However, 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.

Fonts

The PDF indexer must be able to access fonts to insert appropriate information in a PDF output file. If a font is referenced in an input file but is not available on the system, the PDF indexer substitutes a font, usually Courier.

All installations should verify that the standard Adobe font files are installed in the standard font directory. For installations that plan to use the PDF indexer to access double-byte character set (DBCS) fonts, verify the locations of the DBCS font files and export or add the ACRO_RES_DIR and PSRESOURCEPATH environment variables.

Graphical indexer

Since Version 7.1 of the administrative client, it has been possible to define PDF reports within the application component of OnDemand graphically in the same way as line data and AFP. The principle of indexing PDF data is the same as all of the other data types supported by OnDemand; therefore, triggers, fields and indexes must be defined. This section serves as an introduction to the PDF graphical indexer by stepping through an example of indexing a PDF document.

The following process is an extract from “PDF Indexing”, in *IBM Content Manager OnDemand for Multiplatforms Release Notes for Version 7.1.0.10*, which comes with the Content Manager OnDemand for Multiplatforms software. 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. Our intention here is to elaborate on this example by clarifying some of the instructions, and throughout each step, adding important hints, tips, and explanations.

The process is as follows:

1. Start the administrative client.
2. Log on to a server.
3. Start the Report Wizard. Click the **Report Wizard icon** on the toolbar.
4. In the Sample Data window, select **PDF** from the pull-down list of data types, and then click **Select Sample Data**.

5. In the Open window, type the name or full path name of your file in the space provided or use the **Browse** option to locate your PDF file.
6. Click **Open**. The graphical indexer opens the input file in the report window.

If the PDF data fails to view, or an error message such as the one shown in Figure 7-2 is displayed, then you must follow the steps in 7.1.2, “PDF and the OnDemand client” on page 187.

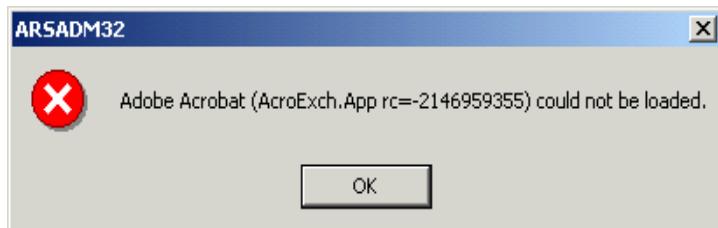


Figure 7-2 Error message if PDF does not view

7. Press the F1 key to open the main help topic for the report window.
8. 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.

Under Options and Commands, click **Indexer Information** page to open the Indexing Commands topic. (You can also use the content help tool to display information about the icons on the toolbar.) Under Tasks, Indexer Information page, click **Adding a trigger (PDF)**.

9. Close any open help topics and return to the report window.
10. Define a trigger as follows:
 - a. Find a text string that uniquely identifies the beginning of a document, for example, Account Number, Invoice Number, Customer Name.
 - b. 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 towards the lower right corner of the string. As you drag the mouse, the graphical indexer uses a dotted line to draw a box. When you have enclose the text string completely inside of a box, release the mouse. The graphical indexer highlights the text string inside of a box.

Important: We recommend that the box that you create around the text string, which you are trying to collect, should be as large as possible to ensure that the field is collected at load time.

Figure 7-3 shows an example of a box that is intended to capture the text string Page 1. 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 Page 1 is shown in the Value entry field, which means that only the Page 1 text 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 have encapsulated *all* of the text string that we must capture.

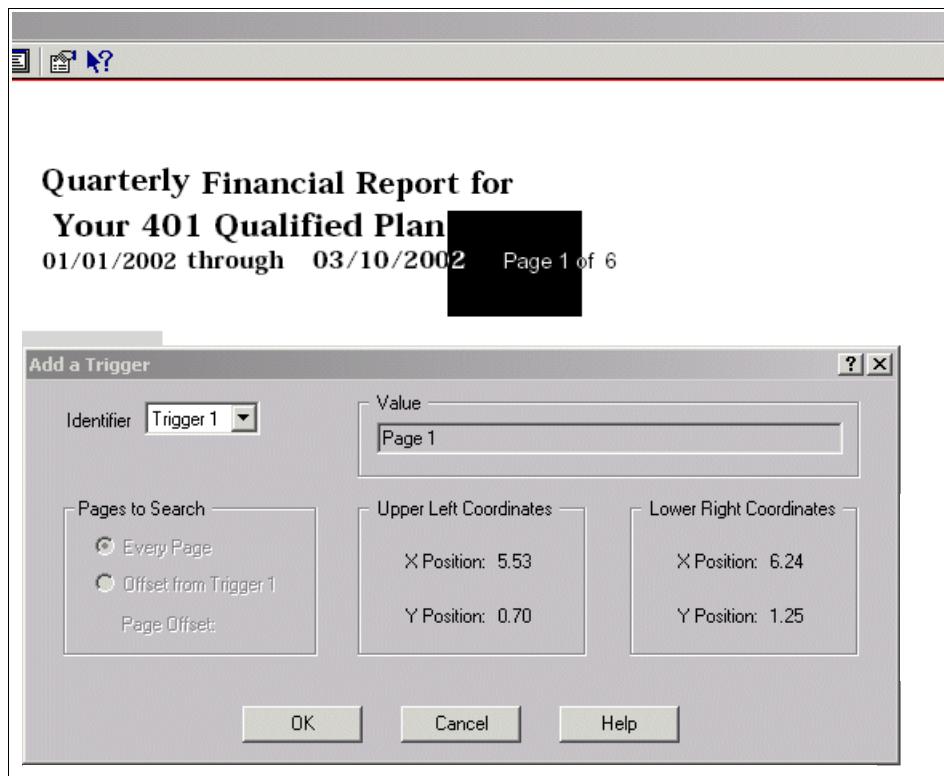


Figure 7-3 Capturing text with the PDF graphical indexer

- 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 the text string in the Value field for Trigger1, is correct. For trigger 1, there are no options or values that you can specify. For other triggers, click **Help** for assistance with the other options and values. Click **OK** to define the trigger.

- e. To verify that the trigger uniquely identifies the beginning of a document:
 - i. 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 should highlight 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. Put the report window in **add** mode.

11. 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 should contain a sample index value. For example, if you want to extract account number values from the input file, then find where the account number is printed on the page.
- b. 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. When you have enclosed the text string completely inside of a box, release the mouse. The graphical indexer highlights the text string inside of a box.

Important: Use exactly the same principles for collecting fields as collecting the trigger text string in step 9 b on page 191. If the fields that need to be collected are close together, we recommend that you 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 these 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 should be displayed under Reference String; the trigger should identify 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 OnDemand to store the index value. In the Folder Field Name field, enter the name of the folder

field to appear on the client search screen. 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:
 - i. Place the report window into display mode. The fields should have a blue box drawn around them.
 - 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 should move to the next document and highlight 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 into add mode.
12. 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 details about the indexing parameters, refer to *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235.
13. When you finish defining all of the triggers, fields, and indexes, press Esc to close the report window.
14. Click **Yes** to save the changes to the indexer parameters.
15. In the Sample Data window, click **Next** to continue with the Report Wizard.

7.3.1 PDF indexing on z/OS

Indexing PDF documents on z/OS uses the same procedure described in the preceding section. However, the load process requires modification of the **arsload** procedure. You have to specify the following items:

- ▶ The Adobe font mapping table (must be created)
- ▶ A font data set that is created by the PDF indexer at runtime to hold the names of the fonts that are located in the font path in the runtime directory
- ▶ A data set that contains the attributes of the temporary working space for the PDF libraries

Creation of the font mapping table

The font mapping table is a z/OS data set where every Adobe Type 1 font is referenced in the system. This mapping table is not shipped with the base OnDemand code. Adobe fonts are not owned by IBM. You must buy the fonts from Adobe and upload them in a binary format to your z/OS system. The PDF indexer requires access to the fonts to insert the appropriate information into the PDF output file. If a font is referenced but not available on the system, it is substituted with a standard font.

The sample delivered in the documentation consists of standard Type 1 Adobe fonts. You can create a sequential file or use a partitioned data set (PDS) to generate the font mapping table. The example created here is a PDS where every member contains one font. Table 7-1 shows the data control block (DCB) parameters for the allocation of the font mapping table.

Table 7-1 Space allocation for font mapping table

Space	LRECL	RECFM	BLKSIZE	DSORG
Cyl 1,1	255	VB	27998	PO

Figure 7-4 shows the sample PDS member list.

TEAM5.PDFLIB.RESOURCE.T1PFA			
Name	Prompt	Size	Created
ARIAL			
ARIALB			
ARIALBI			
ARIALI			
ASANSMM			
ASERIFMM			
COUR			
COURB			
COURBI			
COURI			
INDEX			
SYMBOL			
TIMENR			
TIMENRB			
TIMENRBI			
TIMENRI			
ZAPFDING			
End			

Figure 7-4 Sample PDF font mapping table as PDS

Example 7-1 shows how the ARIAL font is displayed in the PDS member after uploading this font to the z/OS system. All fonts are ASCII character set.

Example 7-1 Arial font member on z/OS

```
%ÜPS-AdobeFont-1.0: ArialMT 001.001.%CreationDate: Wed Jan 27 13:02:03  
1999.%%V
```

Modification of the arsload procedure for PDF indexing

The **arsload** procedure must be modified to archive PDF data. Add the JCL statements in Example 7-2 to the **arsload** procedure or uncomment them if they already exist.

Example 7-2 JCL statements to add for PDF Indexer

```
ADOBERES DD DSN=TEAM5.PDFLIB.RESOURCE.INDEX(ADOBERES),DISP=SHR  
ADOBEFNT DD DSN=TEAM5.PDF405.PLUSP1C.ADOBEFNT.LST,DISP=SHR  
TEMPATTR DD DSN=TEAM5.PDF405.PLUSP1C.TEMPATTR,DISP=SHR
```

ADOBERES DD is used by the OnDemand PDF indexer. It specifies the member that points to the location of the ADOBE font mapping table (the PDS data set that is previously created). Example 7-3 shows the ADOBERES member.

Example 7-3 ADOBERES member

```
*****  
TEAM5.PDFLIB.RESOURCE.INDEX(T1PFA14)  
*****
```

Member T1PFA14 contains the statements as shown in Example 7-4. It can point to any data set containing the ADOBE fonts.

Example 7-4 Location of ADOBE font mapping table

```
PS-Resources-1.0  
FontOutline  
. .  
FontOutline  
ArialMT=TEAM5.PDFLIB.RESOURCE.T1PFA(ARIAL)  
Arial-BoldMT=TEAM5.PDFLIB.RESOURCE.T1PFA(ARIALB)  
Arial-BoldItalicMT=TEAM5.PDFLIB.RESOURCE.T1PFA(ARIALBI)  
Arial-ItalicMT=TEAM5.PDFLIB.RESOURCE.T1PFA(ARIALI)  
Courier=TEAM5.PDFLIB.RESOURCE.T1PFA(COUR)  
Courier-Bold=TEAM5.PDFLIB.RESOURCE.T1PFA(COURB)  
Courier-BoldOblique=TEAM5.PDFLIB.RESOURCE.T1PFA(COURBI)  
Courier-Oblique=TEAM5.PDFLIB.RESOURCE.T1PFA(COURI)  
Symbol=TEAM5.PDFLIB.RESOURCE.T1PFA(SYMBOL)  
TimesNewRomanPS-BoldMT=TEAM5.PDFLIB.RESOURCE.T1PFA(TIMENRB)
```

TimesNewRomanPS-BoldItalicMT=TEAM5.PDFLIB.RESOURCE.T1PFA(TIMENRBI)
TimesNewRomanPS-ItalicMT=TEAM5.PDFLIB.RESOURCE.T1PFA(TIMENRI)
TimesNewRomanPSMT=TEAM5.PDFLIB.RESOURCE.T1PFA(TIMENR)
ZapfDingbats=TEAM5.PDFLIB.RESOURCE.T1PFA(ZAPFDING)
.

The DCB in Table 7-2 is used for this library.

Table 7-2 Data control block used for ADOBERES data set

Space	LRECL	RECFM	BLKSIZE	DSORG
Cyl 1,1	80	FB	8800	PDS

ADOBEFNT DD is used by the OnDemand PDF indexer. It specifies the data set that is used by the OnDemand PDF indexer at run time to hold the names of the fonts that are located in the font path and in the runtime directory.

Important: This file normally does not exist when the PDF indexer is not used before. The indexer does not create it and if you run the **arsload** procedure with a non-catalog data set, it fails with a JCL error. You must create this data set prior running the **arsload** procedure. You cannot just allocate an empty data set because the indexer is looking at the end of the data set and searches for the ASCII End of FILE tag, which is X'0A'. Edit the allocated data set in Hex and put X'0A' at the end on the file.

The DCB in Table 7-3 is used for the ADOBEFNT data set.

Table 7-3 DCB used for the ADOBEFNT data set

Space	LRECL	RECFM	BLKSIZE	DSORG
Cyl 2,2	256	VB	27998	PS

TEMPATTR DD is used by the OnDemand PDF indexer. It specifies the data set that contains the attributes of the temporary working space for the PDF libraries. The content of the data set is as shown in Example 7-5.

Example 7-5 Content of TEMPATTR data set

unit=vio,cyl,spaceround,primary=5,secondary=5,recfm=f,lrecl=13030,blksize=13030

INPUT DD points to the PDF document that must be uploaded to the z/OS system. It can be uploaded as a hierarchical file system (HFS) file or as a z/OS data set. If you want to use it as a z/OS data set, the DCB in Table 7-4 must be used.

Table 7-4 DCB used for the INPUT data set

Space	LRECL	RECFM	BLKSIZE	DSORG
Cyl 5,5	80	FB	27920	PS

If the PDF data set is not transferred correctly to the z/OS system, you might see ABENDS and DUMPS when trying to index it.

Limitations

There are some system limitations that you must consider when using the OnDemand PDF indexer:

- ▶ The PDF indexer cannot process data sets that are greater than 4 GB in size.
- ▶ The PDF indexer supports DBCS languages. Since IBM does not provide any DBCS fonts, you must purchase them from Adobe.
- ▶ Input data delimited with Postscript Pass through markers cannot be indexed.
- ▶ The Adobe Toolkit does not validate link, destinations, or bookmarks to other pages in a document or to other documents. Links or bookmarks might or might not resolve correctly, depending on how you segment your documents.
- ▶ Postscript data generated by applications must be processed by a conversion program such as Acrobat Distiller® before you run the PDF indexer. Acrobat Distiller, however, is not available in a z/OS environment.



OnDemand Web Enablement Kit

In this chapter, we introduce the four common server access points to an OnDemand server via the OnDemand Web Enablement Kit (ODWEK). We discuss how these access points can be used. With sample code and working examples, we demonstrate how integration code can be written in order to customize the access to data stored in OnDemand. We show how Java application programming interface (API)-based portlets give out-of-the-box powerful access to the OnDemand servers.

In this chapter, we cover the following topics:

- ▶ ODWEK architecture
- ▶ Integrating with APIs
- ▶ OnDemand Portlets
- ▶ Deploying the ODWEK servlet

8.1 ODWEK architecture

The OnDemand Web Enablement Kit was designed as a toolkit for Web developers to create browser-based interfaces to OnDemand servers. ODWEK Version 8.3, with the latest level of maintenance applied, provides access to OnDemand servers on all supported server platforms, including UNIX, Linux, Microsoft Windows NT, iSeries, and zSeries.

When the ODWEK code is installed, four options are available for the communication method back to an OnDemand server. In this section, we introduce these access methods, which you should use in conjunction with the following two manuals:

- ▶ *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231
- ▶ *IBM Content Manager OnDemand for Multiplatforms Release Notes for Version 7.1.0.10* (available with the Content Manager OnDemand for Multiplatforms software)

8.1.1 ODWEK access to OnDemand

Using ODWEK, there are four different access points to an OnDemand Server:

- ▶ **Common Gateway Interface (CGI):** Through the arswww.cgi executable
- ▶ **Servlet:** Through the ArsWWWServlet interface
- ▶ **Java API:** By writing your own Java program similar to the ArsWWWServlet but that contains custom functions based on a set of requirements
- ▶ **Portlets:** By using out-of-the-box portlets developed by IBM using Java API

CGI

The CGI implementation of ODWEK is probably the most simple one of the four access points to configure. Refer to “Deploying the CGI program”, in Chapter 2, “Installation and Configuration”, in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231, for details about the files that must be placed within the HTTP Web server environment.

For CGI access to OnDemand, a Web server is required to execute the CGI program, but an application server is not required. An instance of the arswww.cgi program is launched every time a user performs an operation that requires connection to the OnDemand server, such as search and retrieval. Refer to Chapter 6, “Performance” on page 159, to learn about issues associated with this.

Servlet

If a Java servlet implementation of ODWEK is chosen, an HTTP Web server *and* a Java enabled application server are required. Refer to “Deploying the servlet” in Chapter 2, “Installation and Configuration”, in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231, for guidelines about the process to configure and deploy the ODWEK servlet.

Important: The method of deploying a servlet varies greatly between different vendors and even different versions of application servers. Guidance specific to the IBM WebSphere application server is provided in 8.4, “Deploying the ODWEK servlet” on page 225.

The fundamental difference between the servlet and the CGI implementation of ODWEK is that the servlet is managed by an application server, while the CGI is executed by an HTTP Web server. This means that rather than executing many instances of the code and therefore allocating memory for these multiple threads, as in the case of CGI, the servlet runs as a single instance, and memory is managed by the application server. To learn about the issues associated with this, see Chapter 6, “Performance” on page 159.

Java API

The servlet that is supplied with ODWEK after a standard installation of the product is a working example of a servlet that has been written using the Java APIs in order to access OnDemand. When using these APIs with ODWEK, an application server and an HTTP Web server are required if clients require access to OnDemand via a browser. However, unless you are using the Java APIs to write a Web application, a Web server or an application server are not required. For example, if a program is written using these Java APIs for bulk retrieval of documents from OnDemand, you might want to send the objects to a file system without using a Web server or an application server.

Portlets

A portal provides personalized access to a variety of applications and aggregate disparate content sources and services. The OnDemand Portlets developed with Java APIs aggregate OnDemand with other applications. This implementation requires an HTTP Web server, a Java enabled application server, and a portal server.

8.2 Integrating with APIs

Of the four access points to an OnDemand server using the APIs supplied with ODWEK, two (CGI and ArsWWWServlet) are provided with the standard installation code of ODWEK as implementation options. The third access point requires custom coding using the Java APIs. And the fourth is an out-of-the-box solution using Java API developed portlets.

Assuming that you use a browser to view information from OnDemand, there are two places where custom code must be written to integrate with the ODWEK APIs. The first place is the HTML Web pages, which present the user interface, such as logon, search, and hit-list pages; the second place is for a custom servlet that is application specific. Figure 8-1 illustrates the various components within ODWEK that make up the three access points into an OnDemand server and shows the software that is required for each.

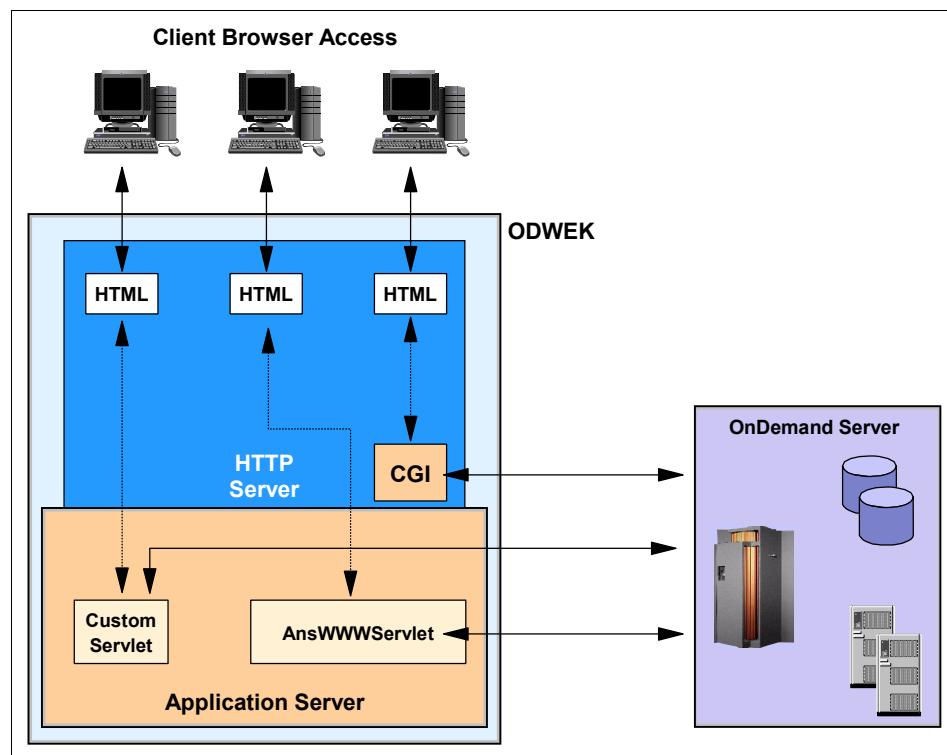


Figure 8-1 ODWEK with CGI, ArsWWWServlet, and a custom servlet

The three HTML documents represent the three access points into the OnDemand server. Figure 8-1 illustrates all three methods of linking to an

OnDemand server. In practice, an implementation of ODWEK only has one of these access methods configured for operation.

8.2.1 HTML samples (URL API)

Samples of the HTML are provided in the samples directory after the standard installation of the ODWEK product. Regardless of the access point (CGI, servlet, or Java API) used to connect to OnDemand, if the user requires the ability to view the data from a browser, then a Web page is required. You must either customize the HTML samples that are provided or simulate the function of these samples via Java script, or as part of a JavaServer™ Page (JSP™) or any standard Web presentation language.

The HTML samples provide basic functions to search and retrieve documents from OnDemand, but they do not show samples of all of the possible APIs that are available for use. The HTML in Example 8-1 on page 204 is a sample of the Uniform Resource Locator (URL) APIs that are used within an HTML document. The HTML is a lengthy example, but demonstrates several uses for the URL APIs.



Figure 8-2 Web page of a sample integrated ODWEK installation

The code shown in Example 8-1 on page 204 is derived from a production application. The server names, IP addresses, and various other sensitive data such as user IDs and password have been removed or altered.

Example 8-1 Sample of the URL APIs used in company.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML ><HEAD><TITLE>Sample Company Intranet</TITLE>
<META content="text/html; charset=windows-1252" http-equiv=Content-Type>
<LINK href=".//Home_files/filelist.xml" rel=File-List>
<META content="MSHTML 5.00.3103.1000" name=GENERATOR>
</HEAD>
<BODY aLink=#009966 bgColor=003399 lang=EN-US
link=#009966 style="tab-interval: 36.0pt" vLink=#009966 onLoad="">
<div id="Layer2" style="position:absolute; left:331px; top:7px; width:128px; height:29px;
z-index:2">
</div>
<div id="Layer3" style="position:absolute; left:125px; top:122px; width:241px; height:180px;
z-index:3"></div>
<div id="Layer4" style="position:absolute; left:404px; top:122px; width:371px; height:134px;
z-index:4">
<p><font size="2" face="Gill Sans" color="#FFFFFF"> </font>
<font size=4> <font size="2" face="Gill Sans" color="#FFFFFF"><b>Company policy<b><br>
<a href="policy.htm"><b>by subject, department, or description</b></a></b></b></font><font
size=4><b><b>
<form action="http://nn.nn.nn.nn/cgi-bin/arswww.cgi method=post>
<font face="Gill Sans" color="#FFFFFF" size="2">
<input name=_folder type=hidden value=policy>
<input name=_display_fields type=hidden value="Policy Subject,Code,Date Revised,Date
Added,Revision">
<input name=_function type=hidden value=dochitlist>
<input name=_max_hits type=hidden value=50>
<input name=_password type=hidden value=XXXXXXX>
<input name=_server type=hidden value=nn.nn.nn.nn>
<input name=_user type=hidden value=XXXXXXX>
<input name=_html type=hidden value=template.htm>
</font><font size="2" face="Gill Sans" color="#FFFFFF">Or type a known
policy code:
<input name="Policy Code" size=6 value="C0322">
and click
<input type=submit value=Submit name="submit">
or press return.</font>
</form>
</b>
<p><font face="Gill Sans" color="#FFFFFF" size="2"><b><a
href="http://nn.nn.nn.nn/policy.nsf/PolicyRequest?OpenForm">
Administration - to archive new policies - restricted access</a></b> </font></p>
</b></font></div>
<div id="Layer6" style="position:absolute; left:9px; top:131px; width:130px; height:95px;
z-index:6"><font face="Gill Sans" color="#FFFFFF" size="2">
</font>
<p></p>
<p><br>
```

```

<font size="2" face="Gill Sans" color="#FFFFFF"><b>Documents<font color="#006699"><b><br>
<a href="//nn.nn.nn.nn/cgi-bin/arswww.cgi?_function=searchcrit&_server=nn.nn.nn.nn&_folder=DocTitle&_user=XXXXXXX&_password=ondemand2&_html=template.htm"><b>by<br>
<a href="//nn.nn.nn.nn/cgi-bin/arswww.cgi?_function=searchcrit&_server=nn.nn.nn.nn&_folder=DocAuth&_or=_user=XXXXXXX&_password=ondemand2&_html=template.htm"><b>by<br>
<a href="//nn.nn.nn.nn/cgi-bin/arswww.cgi?_function=searchcrit&_server=nn.nn.nn.nn&_folder=DocID&_user=XXXXXXX&_password=ondemand2&_html=template.htm"><b>by<br>
<div id="Layer8" style="position:absolute; left:124px; top:53px; width:651px; height:20px; z-index:8"><font face="Gill Sans" color="#FFFFFF" size="2" color="#FFFFFF">Sample Company Intranet.<br><br>
    You can search all internal documents here.<br><b><a href="ondemandhelp.htm">First-time users click here</a></b> as you will need to install viewing software.<![if !supportEmptyParas]>
    </font></div>
<DIV align=center><B> <font color="#FFFFFF" size="5" face="Gill Sans"> </font>
</B> <B> </B> <B>
<P></P>
<font face="Gill Sans" color="#FFFFFF" size="2"><![if !supportEmptyParas]></font></B>
<DIV align=left>
<P><font face="Gill Sans" color="#FFFFFF" size="2"><br>
<br>
</font></P>
<P><font size="2" face="Gill Sans" color="#FFFFFF"><br>
<BR>
</font>
<FONT size=4><FONT size=4><FONT size=4><FONT size=4>
<P><font size="2" face="Gill Sans" color="#FFFFFF"><B><br>
</b></font></P>
<font face="Gill Sans" color="#FFFFFF" size="2">
<SCRIPT LANGUAGE="JavaScript">
<!--
document.forms[0].elements[8].focus();
document.forms[0].elements[8].select();
// -->
</SCRIPT>
</font><font face="Gill Sans" color="#FFFFFF"> </font> </font></font></font></font></DIV>
</DIV></BODY></HTML>

```

To fully complete this code sample, we included the source of the policy.htm file in Example 8-2 on page 206 that is referenced by the HTML sample in Example 8-1 on page 204.

Example 8-2 The policy.htm file referenced by the company.html

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252">
<META NAME="Generator" CONTENT="Notepad">
<TITLE>Company Policy</TITLE>
</HEAD>
<FRAMESET cols="180,572*" rows="*" border="0" framespacing="0" frameborder="NO">
  <FRAME src="http://nn.nn.nn/nopenaccessdbs/policy.nsf/SearchForm?OpenForm" name="sidenav"
scrolling="NO" noresize marginwidth="5" marginheight="10" frameborder="NO">
  <FRAME name="main2" marginwidth="0" scrolling="AUTO" marginheight="0" frameborder="NO"
src="http://nn.nn.nn/nopenaccessdbs/policy.nsf/PolicyCodes?openview&count=15">
</FRAMESET>
<NOFRAMES>
<BODY TEXT="#ffffff" LINK="#ffffff" VLINK="#ffff00" BGCOLOR="#ffffff" alink="#66FF33">

</BODY>
</NOFRAMES>
</HTML>
```

Refer to Chapter 5, “API Reference”, in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231, for a full reference of how to use the URL APIs.

8.2.2 Java API samples

After a standard installation of ODWEK, there is documentation on the Java APIs in the form of a ZIP archive located in the API directory. The HTML files contained within the ZIP file explain usage of the APIs. In this section, we show how the APIs can be used, by presenting working examples of Java code that contain ODWEK Java API calls. We do not show examples of all of the APIs available. However, based on the examples that we have here, you can understand the principles behind using them.

To compile and run these samples, some preliminary work must be done:

- ▶ Ensure that the Java Development Kit is installed as a prerequisite.
- ▶ Ensure that the ODWEK shared library (ARSWWWSL.dll on UNIX and Windows) is accessible:
 - **Windows:** The ODWEK installation directory must be in the system path.
 - **UNIX:** The user running the Java program should have the ODWEK installation directory as part of the PATH variable.

- **OS/390:** In UNIX System Services, the user running the Java program should have the ODWEK installation directory as part of the PATH variable.
- ▶ Ensure that the directory that contains the ODAPI.jar file is in CLASSPATH.
- ▶ The Java API uses the arswww.ini file, although it does not require a Web server to run. Ensure that all referenced paths in the arswww.ini file exist.

Logon and search

The code sample in Example 8-3 is a working example of logging on to an OnDemand server, opening a folder called *Credit Card Statements*, and then searching this folder, followed by a logging off. Also, inside this code sample, such actions as opening the folder and generating the hit list are timed to assess performance and for debugging purposes.

Example 8-3 Code sample of logon and search

```
import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class Search
{
    public static void main (String argv[])
    {
        int rc;
        int numFolders;
        byte[] data;
        String[] displayList;
        FileOutputStream file;
        ODServer odServer;
        ODFolder odFolder;
        ODCriteria odCrit;
        Vector hits;
        ODHit odHit;
        Vector notes;
        Date before, after;
        Date program_start, program_end;

        if (argv.length < 4)
        {
            System.out.println("usage: java Search <server> <user> <pw> <config
dir> [<local server dir>]");
            return;
        }

        try
        {
```

```

program_start = new Date();
odServer = new ODServer();

odServer.initialize(argv[3],
                    "Search.java");

before = new Date();
if (argv.length == 4)
    odServer.logon(argv[0],
                   argv[1],
                   argv[2]);
else if (argv.length == 5)
    odServer.logon(argv[0],
                   argv[1],
                   argv[2],
                   ODConstant.CONNECT_TYPE_LOCAL,
                   0,
                   argv[4]);
after = new Date();
System.out.println("logon: " + (after.getTime() -
before.getTime()));

before = new Date();
if( odServer.getNumFolders() > 0 )
{
    Enumeration g = odServer.getFolderNames();
    do
    {
        System.out.println("folder: "+(String)g.nextElement());
    }
    while(g.hasMoreElements());
}
odFolder = odServer.openFolder("Credit Card Statements");
after = new Date();
odCrit = odFolder.getCriteria("Date");
before = new Date();
hits = odFolder.search();
after = new Date();
//System.out.println("    Number of hits: " + hits.size());
//displayList = odFolder.getDisplayOrder();
//for( int i =0; i < displayList.length; i++)
//{
//    //System.out.println("MAS "+ displayList[i]);
//}
//if (hits.size() > 10000)
//{
//    //for (int i = 0; i < hits.size(); i++)
//{
//        //ODHit odhit = (ODHit)hits.elementAt(i);

```

```

        //String id = odhit.getDocId();
        //System.out.println("MAS DOC ----->id: " + id);
        //System.out.println("MAS DOC ----->type: "+
odhit.getDocType());
        //}
        //}
        odFolder.close();
        odServer.logoff();
        odServer.terminate();
    }
    catch (ODEception e)
    {
        System.out.println("ODEception: " + e);
        System.out.println("  id = " + e.getId());
        System.out.println("  msg = " + e.getMessage());
        e.printStackTrace();
    }
    catch (Exception e2)
    {
        System.out.println("exception: " + e2);
        e2.printStackTrace();
    }
}

```

Bulk document retrieval (search with CALLBACK)

The code sample in Example 8-4 is similar to the logon and search sample shown earlier. In addition, it demonstrates the possibility of retrieving multiple documents from an OnDemand server.

To retrieve a single document from OnDemand, this requires a single search and retrieval from an OnDemand server. If several hundred or several thousand documents must be retrieved, then a single search and retrieve for each document adversely effect performance. If bulk retrieval of documents is required, the CALLBACK API must be used, which means that the documents to be retrieved are collated at the server and then sent back to the custom program as a single operation.

The code in Example 8-4 demonstrates the use of an extended version of the CALLBACK class, which is called *MyCallback*, and it is supplied in Example 8-5 on page 213.

Example 8-4 Code sample of search with CallBack

```

import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

```

```

public
class SearchWithCallback
{
    public static void main (String argv[])
    {
        int rc;
        int numFolders;
        byte[] data;
        FileOutputStream file;
        ODSERVER odServer;
        ODFolder odFolder;
        ODCriteria odCrit;
        Vector hits;
        ODHit odHit;

        if (argv.length < 4)
        {
            System.out.println("usage: java test <server> <user> <pw> <config
dir> [<local server dir>]");
            return;
        }

        try
        {
            odServer = new ODSERVER ();
            odServer.initialize(argv[3],
                               "/servlets/TestServlet");

            if (argv.length == 4)
                odServer.logon(argv[0],
                               argv[1],
                               argv[2]);
            else if (argv.length == 5)
                odServer.logon(argv[0],
                               argv[1],
                               argv[2],
                               ODConstant.CONNECT_TYPE_LOCAL,
                               0,
                               argv[4]);

            numFolders = odServer.getNumFolders("C%");
            System.out.println("number of folders is: " + numFolders);

            odFolder = odServer.openFolder("Credit Card Statements");
            odCrit = odFolder.getCriteria("Account");
            //odCrit.setOperand(ODConstant.OPBetween);
            odCrit.setOperand(ODConstant.OPLike);
        }
    }
}

```

```

odCrit.setSearchValue("%");

//ODCallback odc = new ODCallback();
MyCallback odc = new MyCallback();
//odc = odFolder.searchWithCallback();
//odFolder.searchWithCallback("where account LIKE '%'", odc);
String sql = "where account LIKE '%'";
odFolder.searchWithCallback(sql,
                           "",
                           "",
                           odc);

// wait for the query to finish
//odc.waitForOperation();

// allow the user to cancel
FileInputStream fis = new FileInputStream(FileDescriptor.in);
int i = 0;
System.out.println("enter the number 1 to stop search:");
while (!odc.isDone())
{
    if (fis.available() != 0)
        i = fis.read();
    if (i == 0x31)
    {
        System.out.println("cancelling the search...");
        odc.cancel();
        System.out.println("search cancelled");
    }
    else
        Thread.sleep(100);
}
System.out.println();
System.out.println("done searching");
hits = odFolder.getHits();

System.out.println("Number of hits: " + hits.size());
if (hits.size() != 0)
{
    // String[] displayOrder = odFolder.getDisplayOrder();
    // for (int i = 0; i < displayOrder.length; i++)
    // {
    //     System.out.print(displayOrder[i] + "\t\t");
    // }
    // System.out.println();
    // for (int j = 0; j < hits.size(); j++)
    // {
    //     odHit = (ODHit)hits.elementAt(j);
}

```

```

        //           for (Enumeration e = odHit.getDisplayValues();
        //               e.hasMoreElements();
        //               )
        //               {
        //                   System.out.print((String)e.nextElement() + "\t\t");
        //               }
        //               System.out.println();
        //           }
        }

        odServer.terminate();
    }
    catch (ODEException e)
    {
        System.out.println("ODEException: " + e);
        e.printStackTrace();
    }
    catch (Exception e2)
    {
        System.out.println("exception: " + e2);
        e2.printStackTrace();
    }
}

static String getOpName(int op)
{
    String s;

    switch (op)
    {
        case ODConstant.OPEqual:
            s = "Equal";
            break;
        case ODConstant.OPNotEqual:
            s = "Not Equal";
            break;
        case ODConstant.OPLessThan:
            s = "Less Than";
            break;
        case ODConstant.OPLessThanEqual:
            s = "Less Than or Equal";
            break;
        case ODConstant.OPGreaterThan:
            s = "Greater Than";
            break;
        case ODConstant.OPGreaterThanOrEqual:
            s = "Greater Than or Equal";
            break;
        case ODConstant.OPIn:
    }
}

```

```

        s = "In";
        break;
    case ODConstant.OPNotIn:
        s = "Not In";
        break;
    case ODConstant.OPLike:
        s = "Like";
        break;
    case ODConstant.OPNotLike:
        s = "Not Like";
        break;
    case ODConstant.OPBetween:
        s = "Between";
        break;
    case ODConstant.OPNotBetween:
        s = "Not Between";
        break;
    default:
        s = "Operator unknown";
        break;
    }

    return s;
}

```

CALLBACK

The code sample in Example 8-5 extends the CALLBACK class and is used in the sample code that is shown in Example 8-4 on page 209.

Example 8-5 Code sample of CALLBACK

```

import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class MyCallback extends ODCallback
{
    MyCallback(ODFolder folder)
    {
        m_folder = folder;
    }

    public void HitHandleCallback(int hit, int off, int len)
    {
        System.out.println("hit: " + hit + ", off=" + off + " len=" + len);
    }

```

```

public boolean DataCallback(byte[] data)
{
    System.out.println("data length: " + data.length);
    return true;
}

public boolean HitCallback(String docid,
                           char type,
                           String[] values)
throws Exception
{
    System.out.println("id " + docid + ": " + type + " ");
    return true;
}

ODFolder m_folder = null;
}

```

UPDATE

The code in Example 8-6 demonstrates the use of the UPDATE API. In this sample, we see a logon, a search, and then use of the UPDATE API. It is possible to alter the index information for the hits that are returned from the search.

Example 8-6 Code sample of updating a document index

```

import java.util.*;
import java.io.*;
import com.ibm.edms.od.*;

public class Logon
{
    public static void main (String argv[])
    {
        int      rc;
        int      numFolders;
        String   info;
        String   fldname;
        ODServer odServer;
        ODCriteria odcrit;
        ODFolder odfolder;
        ODHit   odhit;
        byte[]   data = null;
        byte[]   data2;
        Vector   hits = new Vector();
        Hashtable hshApprovalVals = null;

        if (argv.length < 4)

```

```

{
    System.out.println("usage: java Logon <server> <user> <pw> <config dir>
[<local server dir>]");
    return;
}

try
{
    odServer = new ODServer ();
    System.out.println("calling initialize with "+argv[3]);
    odServer.initialize(argv[3],
                        "Logon.java");
    System.out.println("Did the Initialize");
    odServer.setServer(argv[0]);
    odServer.setUserId(argv[1]);
    odServer.setPassword(argv[2]);
    odServer.setPort(1445);
    odServer.logon();
    System.out.println("Did a Logon");
    odfolder = odServer.openFolder("Credit Card Statements");
    odcrit = odfolder.getCriteria("Account");
    System.out.println("Open Folder");
    odcrit.setSearchValue("000-000-000");
    odcrit.setOperand(ODConstant.OPEqual);
    hits = odfolder.search();
    System.out.println("Got Hits");
    odhit = (ODHit)hits.elementAt(0);
    System.out.println("Got odhit");
    info = odhit.getDocId();

    System.out.println("Information is "+info);

    hshApprovalVals = new Hashtable();

    System.out.println("Created new hash table");
    hshApprovalVals.put("Account", "100-000-000");

    System.out.println("Put values in the hash table");

    odhit.update(hshApprovalVals);

    System.out.println("Updated the hit");

    odServer.logoff();

    System.out.println("Logged off");
    odServer.terminate();
    System.out.println("Terminated the server object");
}

```

```
        }
        catch (ODEException e)
        {
            System.out.println("ODEException: " + e);
            System.out.println("  msg = " + e.getErrorMsg());
            System.out.println("  msg = " + e.getErrorCode());
        }
        catch (Exception e2)
        {
            System.out.println("exception: " + e2);
            e2.printStackTrace();
        }
    }
}
```

8.3 OnDemand Portlets

The IBM Portlets V3.2 for IBM DB2 Content Manager OnDemand for Multiplatforms V8.3 is a new release. The OnDemand Portlets provide a portlet-based client interface to work with OnDemand servers. The OnDemand Portlets are built using the ODWEK.

The portlets are available in the IBM Workplace™ Solutions catalog at the following Web address:

<http://catalog.lotus.com/wps/portal/portlet/catalog>

8.3.1 Features and functions included in the OnDemand Portlets

Some of the features and functions included in the OnDemand Portlets are:

- ▶ Connect to the OnDemand backend server
- ▶ Change expired password
- ▶ Folder list with sort and pagination support
- ▶ Multiple predicate search, any or all search
- ▶ Default values and fixed values support in search criteria
- ▶ Sort and pagination support for search results
- ▶ Annotation status on search results
- ▶ View or Append Notes support
- ▶ Server print documents
- ▶ Local print annotations

- ▶ Multiple Portlets: Main and Viewer portlets
- ▶ Line Data applet support
- ▶ AFP2HTML applet support
- ▶ Large Object support for Advanced Function Presentation (AFP) Plug-in, Line Data Applet, and AFP2HTML Applet
- ▶ Configuration support through the arswww.ini file
- ▶ Logout from the OnDemand server

Two portlets are delivered in OnDemand Portlets: Main and Viewer portlets.

The *Main portlet* provides a single portlet-based functional equivalence to the Content Manager eClient application when accessing an OnDemand server. The Main portlet has a serial organization of interfaces so that only a single interface panel is displayed at a time. When using only the Main portlet, documents are displayed in new browser windows.

The *Viewer portlet* can interact with the Main portlet to display documents on the same portal page instead of in new browser windows. Both the Main and Viewer portlets must be on the same portal page. Several types of viewers can be used to view the documents based on the ODWEK configuration.

Note: The OnDemand Portlets are national language support (NLS) ready, but for now only the English language is available.

8.3.2 Hardware and software requirements

In this section, we cover the hardware and software requirements, as well as the supported OnDemand backend server platforms.

Hardware requirements

The OnDemand Portlets are supported on Windows, AIX, Solaris, and Linux. The portlets have been tested with WebSphere Portal 5.1 on the following platforms:

- ▶ Windows 2000 and Windows 2003
- ▶ AIX 5.2
- ▶ Solaris 9.0
- ▶ Linux RHEL 3 and RHEL 4

Software requirements

The OnDemand Portlets require that the following software products are installed on the *Portal Server machine* where they will be deployed.

- ▶ WebSphere Application Server Version 5.1.1 or later
- ▶ WebSphere Business Integration Server Foundation Version 5.1.1
- ▶ WebSphere Portal Enable for Multiplatform Version 5.1.0.1
- ▶ IBM DB2 Content Manager OnDemand Web Enablement Kit 7.1.2.5

The *client system* must be capable of rendering HTML generated by WebSphere Portal Enable Version 5.1. Most modern browsers satisfy this requirement.

Supported OnDemand backend server platforms

The OnDemand Portlets use ODWEK to connect to the backend OnDemand servers. The OnDemand Portlets can be used as a client to any OnDemand server that is supported by the ODWEK V8.3 (also known as V7.1.2.5).

The OnDemand Portlets have been tested against OnDemand servers running on the following platforms:

- ▶ OnDemand V7.1 for Solaris
- ▶ OnDemand V7.1 for z/OS
- ▶ OnDemand for i5 Common Server
- ▶ OnDemand V7.1 for AIX
- ▶ OnDemand V7.1 for Windows
- ▶ OnDemand V7.1 for Linux (Red Hat and SUSE)

8.3.3 Configuring and deploying the IBM OnDemand Portlets 3.2

In this section, we explain the steps to install and deploy the OnDemand Portlets on a WebSphere Portal server.

1. Verify that all the prerequisite software is installed and validated successfully.
2. Configure the WebSphere Portal Server.
3. Configure the shared library path environment variable for ODWEK.
4. Install (or update) the OnDemand Portlets.
5. Complete the OnDemand Portlets configuration.

In this section, we concentrate on step 5, the configuration. For more details about other steps, refer to the OnDemand Portlets readme file, which you download from the catalog with OnDemand Portlets as explained in 8.3, “OnDemand Portlets” on page 216.

OnDemand Portlets configuration

The configuration of the OnDemand Portlets is done consistently with the current ODWEK configuration. The portal administrator must modify the entries in the *arswww.ini* configuration file delivered with ODWEK. You can add new configuration options for the IBM OnDemand Portlets as defined in the [PORTLET CONFIGURATION] section to customize the use of the OnDemand Portlets. You can also update the existing ODWEK configuration options as appropriate to customize the behavior of the OnDemand Portlets.

Configuring the arswww.ini file

The configuration must be done for the *arswww.ini* file at least *once* for every ODWEK installation with which the OnDemand Portlets will be associated with.

In the *arswww.ini* file, the new section, [PORTLETS CONFIGURATION], is introduced specifically for use with the OnDemand Portlets. The parameters within this section provide the configuration options that affect the user interface of the OnDemand Portlets. This section is optional.

The [PORTLETS CONFIGURATION] section contains the following parameters:

- ▶ **FOLDERSPERPAGE**

The value of this parameter is used to determine the number of folders to be displayed per page in the Folder List panel. If this parameter is not specified, the default used is 25 folders per page. A value of “-1” indicates that all folders will be displayed on one page.

`FOLDERSPERPAGE=<count>`

- ▶ **HITSPERPAGE**

The value of this parameter is used to determine the number of hits to be displayed per page in the Search Results panel. If this parameter is not specified, the default used is 25 hits per page. A value of “-1” indicates that the entire hit list is displayed in the Search Results panel. No page navigation controls appear on the Search Results panel and a scroll bar is displayed for navigating the list. This parameter is optional. See also the MAXHITS parameter, which restricts the number of hits retrieved by the ODWEK API. The OnDemand Portlets are only able to display up to the MAXHITS parameter, if specified.

`HITSPERPAGE=<count>`

► **LAUNCHPRINTDIALOG**

The value of this parameter determines if the Browser Print window must be displayed when in Print is invoked on the View Notes panel to print the notes for the selected document. If not specified, the Print window is launched along with the Print Preview window.

LAUNCHPRINTDIALOG= {0| 1}

► **VIEWWITHVIEWERPORTLET**

The value of this parameter determines if a document must be viewed with the viewer portlet. The default is 1, which means that the documents are viewed with the viewer portlet. If disabled, individual browser windows are launched to view the documents. This parameter is optional. If not specified, the viewer portlet is used to view the documents.

VIEWWITHVIEWERPORTLET= {0|1}

► **LOGLEVEL**

The value of this parameter determines the log level used in the OnDemand Portlets.

LOGLEVEL= [DISABLE | DEBUG | ERROR | INFO]

Note the following explanation:

- DISABLE: No logging
- ERROR: Unsuccessful completion of operations
- INFO: Successful completion of operations
- DEBUG: Debug or trace messages

The default value is DEBUG. The DEBUG level includes all DEBUG, INFO, and ERROR level messages. The INFO level includes INFO and ERROR level messages. The ERROR level includes only the ERROR level messages. Choosing DISABLE does not log any messages.

► **LOGFILEPATH**

The value of this parameter specifies the absolute path of the log file used to generate the OnDemand Portlets logs.

LOGFILEPATH=<full path name>

Consider the following example:

LOGFILEPATH=c:\\temp\\odpTrace.log

8.3.4 Using OnDemand Portlets

The OnDemand Portlets provide the view mode and help mode. There is no credential vault support in this release. Each panel within the portlet has an associated help topic that provides guidance for that panel.

There are several ways to view documents with the OnDemand Portlets based on viewer configuration. Using the OnDemand Portlets specific configuration, the viewer portlet can be used for viewing multiple documents in a tabbed pane, or the main portlet can be used to launch new browser windows. The type of viewer to be used for a given document type is completely controlled by the ODWEK configuration. For more information about the ODWEK related viewing configuration and viewing engines, refer to *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231.

The following viewers are available for the OnDemand Portlets:

- ▶ **AFP Plug-in Viewer**

This viewer displays AFP documents typically stored in an OnDemand server. It is optimized to display large documents.

- ▶ **AFP2HTML Applet Viewer**

This viewer displays AFP documents as HTML pages. It can be enabled if the AFP2WEB transform is installed and configured on the server.

- ▶ **Line Data Applet Viewer**

This viewer displays line data documents typically stored in an OnDemand server. It is optimized to display large, textual, and columnar documents.

- ▶ **The Browser Viewer**

This viewer displays the document in a Web browser without any conversion. The document is displayed based on how the different file types are defined for the Web browser. It launches the defined application to display the document with necessary plug-ins.

OnDemand Portlets interface

In the following sections, you see sample displays of the OnDemand Portlet interface. Specifically, you see:

- ▶ Logon panel
- ▶ Folder list
- ▶ Search panel
- ▶ Hit list
- ▶ Image document displayed by the Viewer portlet
- ▶ AFP document displayed by the Viewer portlet
- ▶ Line data document displayed by the Viewer portlet

Logon panel

The Main Portlet provides a logon interface with the OnDemand server list according to the arswww.ini file. See Figure 8-3.

The screenshot shows the 'IBM DB2 Content Manager OnDemand V8.3 Main' window. On the left, there is a 'Server:' dropdown menu set to 'ODServer', a 'User ID:' input field containing 'demo1', and a 'Password:' input field containing '*****'. Below these fields is a 'Log in' button. To the right, a message box displays 'No documents are selected to view.'

Figure 8-3 Logon panel in the OnDemand Portlet

Folder list

After a successful logon, you see a list of the Folders that you are authorized to view. See Figure 8-4. In this example, according to the FOLDERPERPAGE parameter in the arswww.ini, ten folder names are shown per page. There are total 26 folders, but only 10 are shown here. You can sort the list can by either folder name or folder description.

The screenshot shows the 'IBM DB2 Content Manager OnDemand V8.3 Main' window with a 'Logout demo1@ODServer - OD' link at the top. Below it, a 'Folder List' section is displayed. The table header includes 'Folder' and 'Description'. The data rows are:

Folder	Description
AccountDetail	Gas and Electric
Adobe PDF	Demo Folder PostScript MIC
BlueBell Telephone	Telephone
CheckDemoNet	
Cheque_Images	Cheque Images
Credit Card Statements	Bank A - AFP
Customer_Information	Demo Folder
FinanceCorpChecks	Fin@nce Corp. Checks
FinanceCorpTrans	Fin@nce Corp. Monthly Activity Report
Insurance_Policies	Insurance Policies

Figure 8-4 Folder list in the OnDemand Portlet

Search panel

The Folder search panel provides a list with all the previously accessed folders. The search field attributes and the predicate options are carried forward to this panel from the Folder definition in OnDemand. See Figure 8-5.

The screenshot shows the OnDemand search interface. At the top, it says "IBM DB2 Content Manager OnDemand V8.3 Main". Below that is a breadcrumb trail: "Folder List > Search Customer Information". To the right of the trail is a "Logout" link and the session information "demo1@ODServer - OD". A message on the right side states "No documents are selected". The main area is titled "Customer Information". It contains four search fields: "Customer Name" (Like), "Account" (Like), "Documents" (Equal), and "Date" (Between, set to 01/01/04 and 12/31/04). Below these fields are two radio buttons: "ALL of these terms" (selected) and "ANY of these terms". At the bottom are "Search" and "Reset" buttons.

Figure 8-5 Search panel in the OnDemand Portlet

Hit list

After a search is performed, a hit list is displayed. See Figure 8-6. The hit list can be sorted by any of the indexes. The number of hits per page is controlled by the HITSPERPAGE parameter in the arswww.ini file.

The screenshot shows the search results page. The breadcrumb trail is "Folder List > Search Customer Information > Search Results". The main area is titled "Customer Information". It displays a table of search results with columns: "Customer Name", "Account", "Document", and "Date". There are five rows, each representing a document found under the account "PIANO Co". The table includes navigation links at the top and bottom: "Showing 1 - 5 of 5", "Page 1 of 1", "Jump to page", and a page number input field.

Customer Name	Account	Document	Date
PIANO Co	000-000-152	Customer Information	11/15/04
PIANO Co	000-000-152	Customer Information	10/20/04
PIANO Co	000-000-152	Customer Information	11/30/04
PIANO Co	000-000-152	Customer Information	11/21/04
PIANO Co	000-000-152	Customer Information	11/15/04

Figure 8-6 Hit list in the OnDemand Portlet

Imaged document displayed by the Viewer portlet

You can view a document from the hit list. The document is displayed in the Viewer portlet. See Figure 8-7 for an image of the document that is displayed by the OnDemand Viewer portlet.

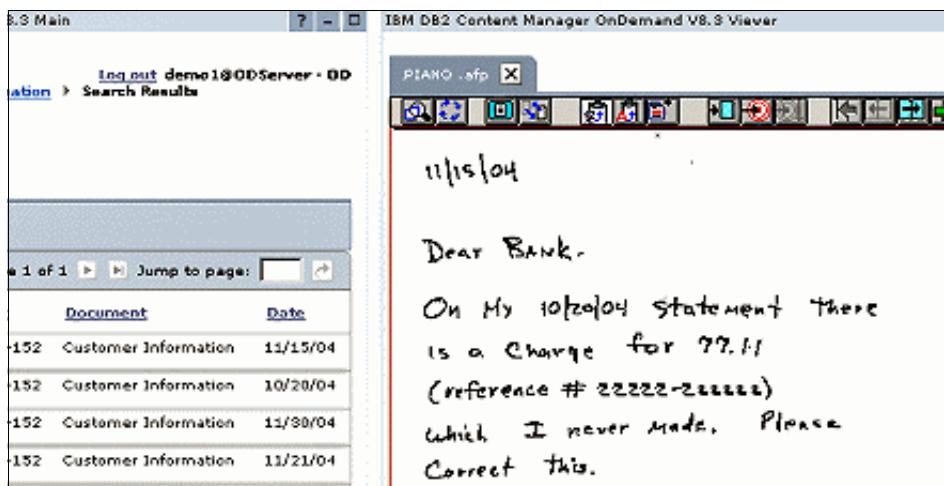


Figure 8-7 Image document displayed by the OnDemand Viewer portlet

AFP document displayed by the Viewer portlet

You can also view an AFP document. See Figure 8-8 for an AFP document that is displayed by the Viewer Portlet. Tabs across the top of the viewer window allow you to quickly jump from one document to another.

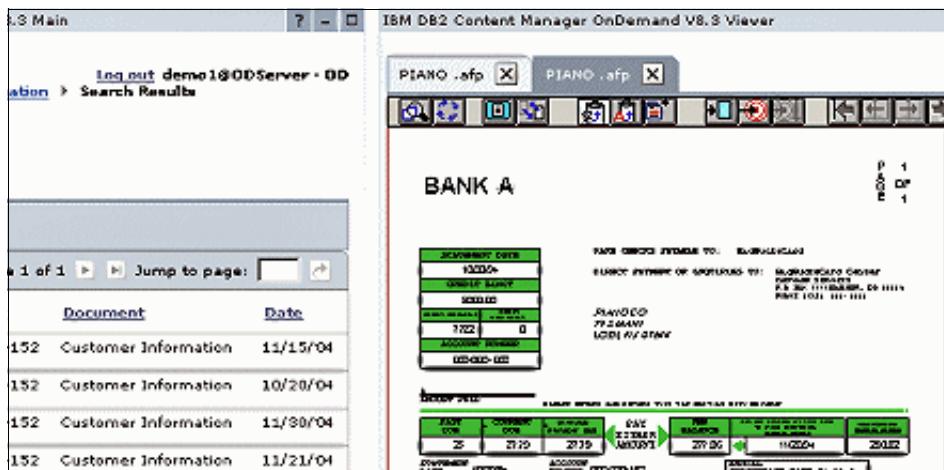


Figure 8-8 AFP document displayed by the OnDemand Viewer portlet

Line data document displayed by the Viewer portlet

In Figure 8-9, three documents are now available in the Viewer Portlet. On the top is the line data document. Tabs allow you to quick jump from one document to another, the image, AFP, and the line documents. Other document types, such as the Portable Document Format (PDF), work the same way.

The screenshot shows the IBM DB2 Content Manager OnDemand V8.3 Viewer interface. At the top, there are tabs for 'PIANO .afp', 'PIANO .afp', and 'PIANO .lin'. The main content area displays a line data document titled 'REPORT CUST152 BANK A'. The document header reads: 'SUMMARY OF CREDIT CARD CHARGES FOR CUSTOMER 000-000-000 FOR TIME PERIOD 01/01/2005 THROUGH 09/20/2005 SORTED'. Below this is a table with columns 'DATE' and 'TRANSACTION DESCRIPTION'. The data in the table is as follows:

DATE	TRANSACTION DESCRIPTION		
01/20/05	WAREHOUSE A	WESTMINSTER	CO
01/20/05	AIRLINES T	ARVADA	CO
01/20/05	WAREHOUSE A	DENVER	CO
01/20/05	WAREHOUSE T	PETOSKEY	MI
01/20/05	AIRLINES	DENVER	CO
01/20/05	TRAVEL	LOUISVILLE	KY

Figure 8-9 Line data document displayed in OnDemand Portlet

8.4 Deploying the ODWEK servlet

The ODWEK servlet can be deployed on application servers that support Java servlets. The method for deploying servlets varies greatly not only between different vendors, but also between different versions and releases of the same product. This section deals with deploying the ODWEK servlet within the environment of the IBM WebSphere Application Server Version 5.1.

This section updates the *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231. It supersedes Chapter 6, "Deploying the Java Servlet" and the following sections:

- ▶ Section 1, "Before You Begin"
- ▶ Section 2, "Copying Files"
- ▶ Section 3, "Deploying the servlet using WebSphere Tools" and subsection 1, "Assembling the Application"

It does not update Section 3, “Deploying the servlet using WebSphere Tools”, nor Subsection 2, “Installing the Application”. For more information, refer to the IBM WebSphere Information Center at the following Web address:

<http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp>

When you reach this Web address, search the topic ID “trun_appl”.

8.4.1 Before you begin

Before you begin deploying the servlet, you must ensure that you comply with the following requirements:

- ▶ Have completed the software installation
See Chapter 4, “Installing ODWEK,” in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231.
- ▶ Have the current version of the IBM HTTP server installed, configured, and operating on the system
- ▶ Have the current version of the IBM WebSphere Application Server installed, configured, and operating on the system

We recommend that you use the WebSphere tools to deploy the servlet. The WebSphere tools automatically configure the HTTP server and Web application server configuration files. If you are an experienced Web server administrator, you might choose not to use the WebSphere tools and deploy the servlet by manually configuring the HTTP server and Web application server configuration files.

To use the WebSphere tools to deploy the servlet, follow these steps:

1. Copy the files.
2. Deploy the servlet using the WebSphere tools.

8.4.2 Copying the files

You must copy the files by completing the following steps:

1. Copy the *ArsWWWInterface.class* file to a directory that is set by the CLASSPATH variable for the Web application server. Because this file is part of a package (com.ibm.edms.od), mirror the package structure as subdirectories under this directory. For example, if the directory *server_root/classes* is set by the CLASSPATH variable, then you must copy the *ArsWWWInterface.class* file to the *server_root/classes/com/ibm/edms/od* directory.

2. Copy the shared library to a directory that is set by the shared library path variable. See Table 8-1.

Table 8-1 Copy of the shared library according to the operating system

Operating system	Shared library path variable	Shared library
AIX	LIBPATH	libarswwwsl.a
HP-UX	SHLIB_PATH	libarswwwsl.sl
Linux	LD_LIBRARY_PATH	libarswwwsl.so
Solaris	LD_LIBRARY_PATH	libarswwwsl.so
Windows	PATH	arswwwsl.dll

3. For Windows systems, copy these files to the directory in which you copied the shared library:
 - ARSSCKNT.DLL
 - ARSCT32.DLL
4. Copy the following files to the HTTP server directory. See Table 8-2.
 - ARSWWW.INI
 - AFP2HTML.INI
 - AFP2PDF.INI

Table 8-2 Copy of the ODWEK INI files according to the operating system

Operating system	HTTP server directory
AIX	/usr/lpp/IBM HTTP Server/bin
HP-UX	/opt/IBM HTTP Server/bin
Linux	/opt/IBM HTTP Server/bin
Solaris	/opt/IBM HTTP Server/bin
Windows	C:\IBM HTTP Server\bin

8.4.3 Deploying the servlet using WebSphere tools

We recommend that you deploy the servlet using the WebSphere tools. The WebSphere tools can perform all of the tasks required to deploy the servlet.

There are two steps in deploying the servlet using the WebSphere tools:

1. Assemble the application with the WebSphere Application Assemble Tool.

2. Install the application from the WebSphere administration console. See “Installing the application”, in *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231.

Assembling the application

To assemble the application with the WebSphere Application Assemble Tool, follow these steps:

1. Start the WebSphere Application Server Toolkit. See Table 8-3.

The WebSphere Application Server Toolkit is available on CD-ROM in the IBM WebSphere Application Server package as a separate installation.

Table 8-3 WebSphere Application Server Toolkit start command by operating system

Operating system	Start command
AIX	ASTK_install_root/astk
HP-UX	ASTK_install_root/astk
Linux	ASTK_install_root/astk
Solaris	ASTK_install_root/astk
Windows	Start → Programs → IBM → ASTK → ASTK

2. You might be prompted to specify a workspace location if a default has not been configured. Select a location and click **OK**.

3. The workbench opens and should contain no projects if the workspace is empty. See Figure 8-10. Select **File** → **New** → **Project**.

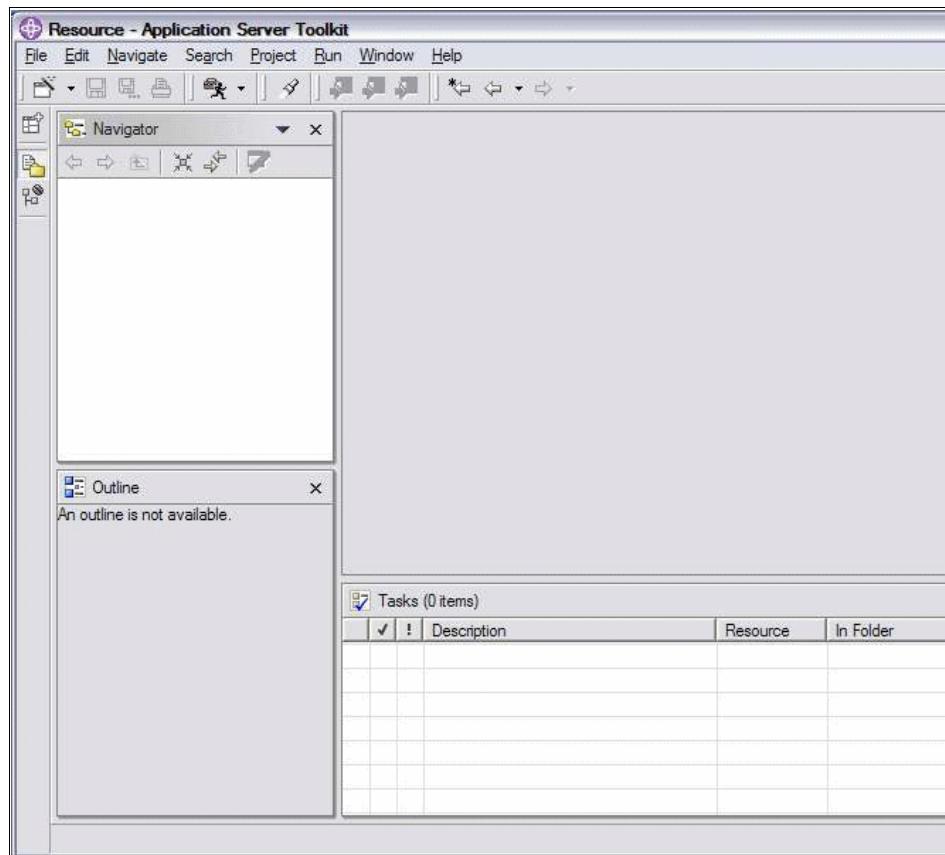


Figure 8-10 WebSphere Application Server Toolkit workbench

4. In the New Project window (Figure 8-11), select **Enterprise Application Project** and click **Next**.

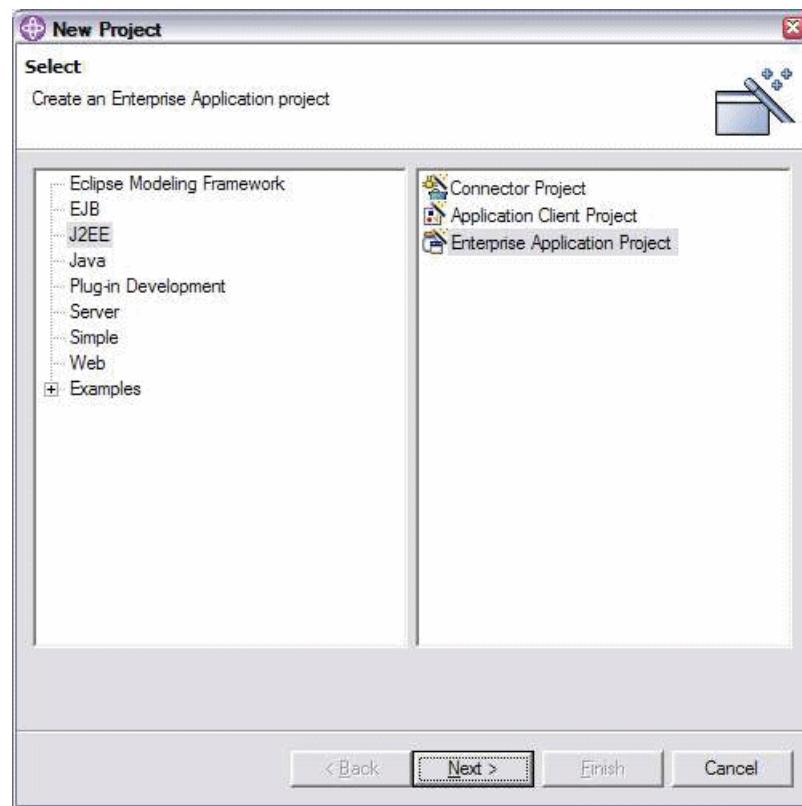


Figure 8-11 Creating a new project

5. In the J2EE Specification version panel (Figure 8-12), select **Create J2EE 1.3 Enterprise Application project** and click **Next**.

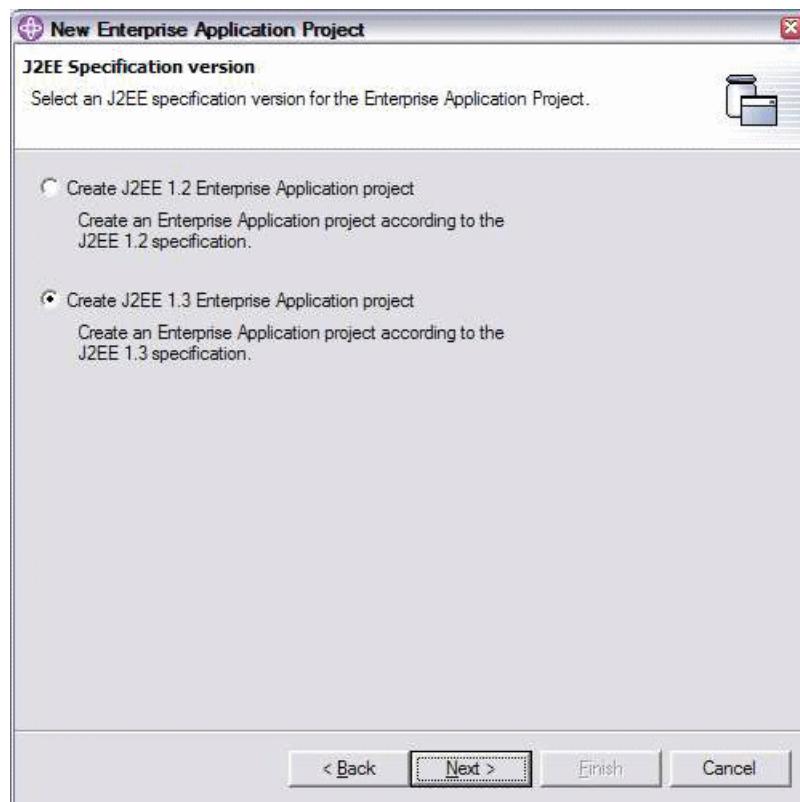


Figure 8-12 J2EE Specification version panel

6. In the Enterprise Application Project panel (Figure 8-13), enter a project name, for example, OnDemandWEK. For Target server, select **WebSphere Application Server v5.1**, which is the target server level that matches the WebSphere Application Server to which you will deploy. Click **Next**.

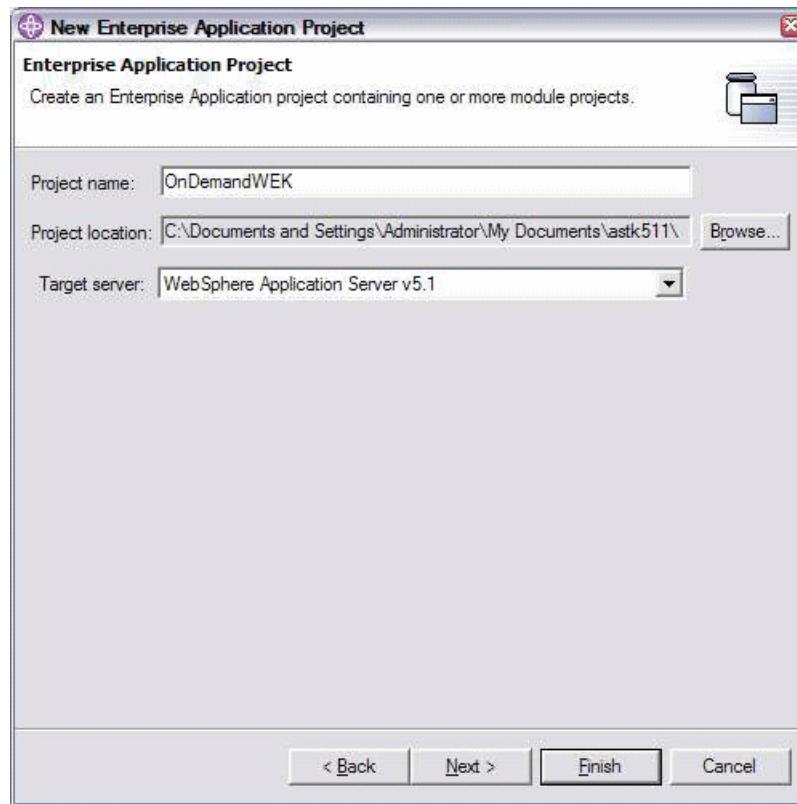


Figure 8-13 Enterprise Application Project panel

7. In the EAR Module Projects panel (Figure 8-14), click **New Module**.

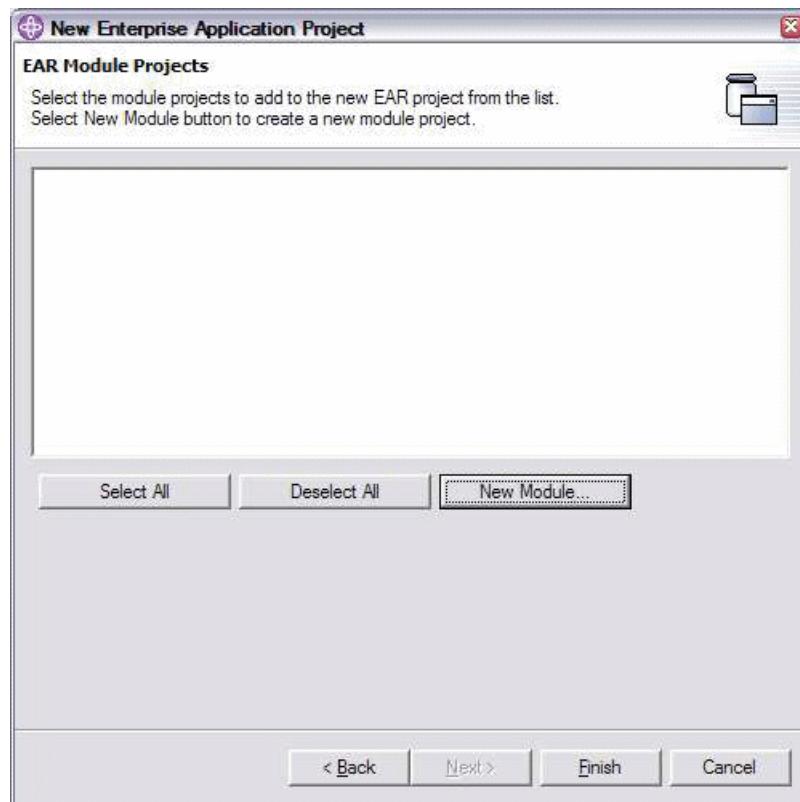


Figure 8-14 EAR Module Projects panel

8. In the New Module Project window (Figure 8-15), leave **Create default module project** selected. Then, from the remaining options, select only the **Web Project** module. Click **Finish**.

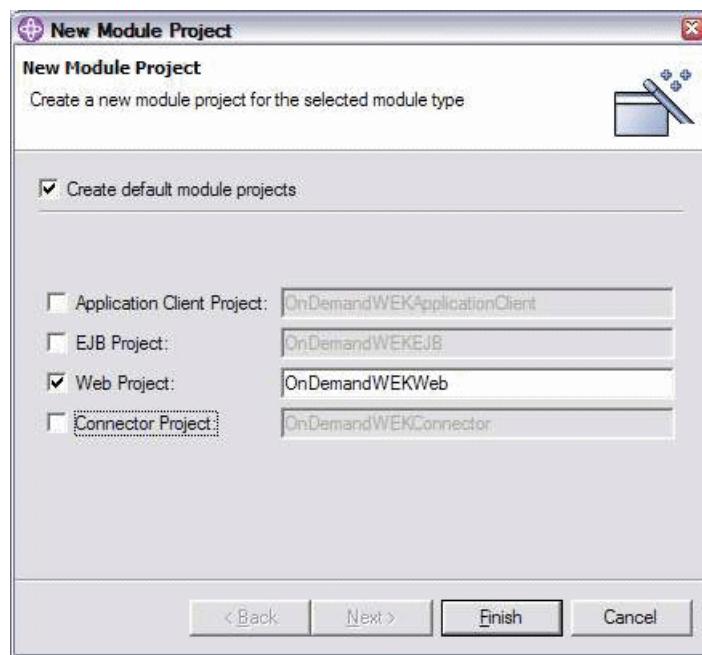


Figure 8-15 New Module Project panel

9. You return to the EAR Module Projects panel (Figure 8-16). Click **Finish**.

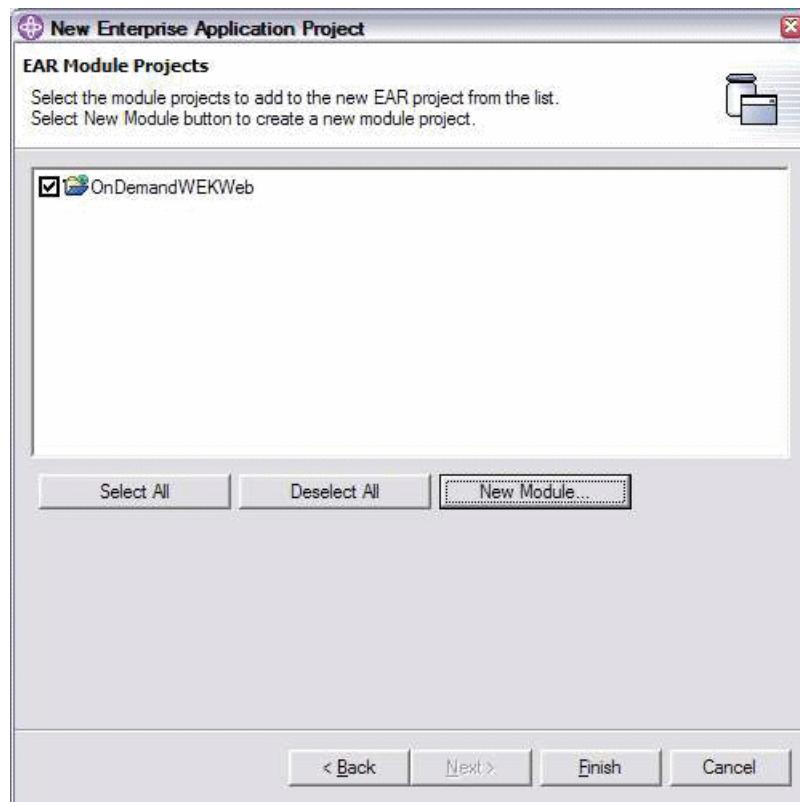


Figure 8-16 EAR Module Projects panel

10. The system prompts you if you want to switch to the J2EE perspective. See Figure 8-17. Click **Yes**.

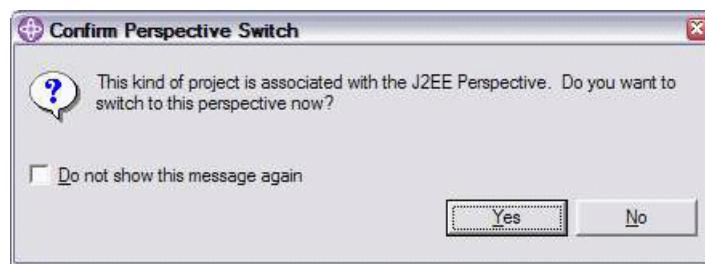


Figure 8-17 Confirm Perspective Switch window

11. Enter the Application Server Toolkit's J2EE perspective. Expand the Enterprise Applications and Web Modules project folders. Two new projects should exist that reflect both the Web project (OnDemandWEKWeb) and its associated enterprise application project (OnDemandWEK). See Figure 8-18.

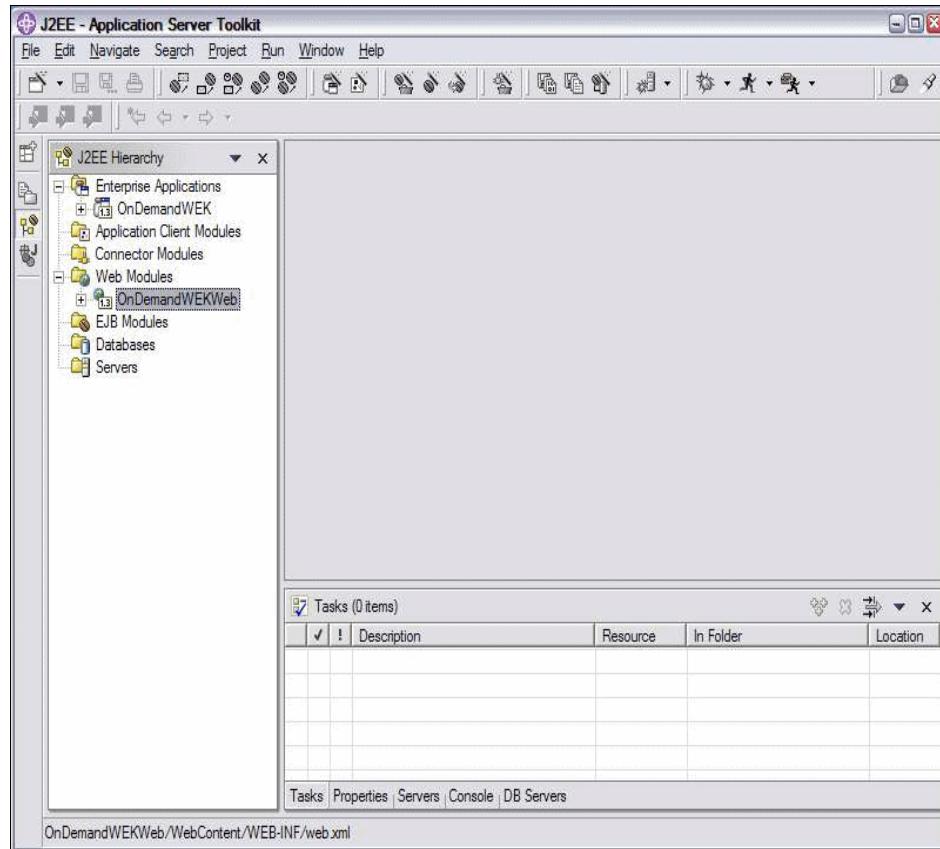


Figure 8-18 WebSphere Application Server Toolkit J2EE perspective

- 12.Right-click the **OnDemandWEKWeb** Web module and select **Import...** → **Import Class Files**
- 13.In the Import Class Files panel (Figure 8-19), select **Import from Zip or Jar**. Click **Next**.

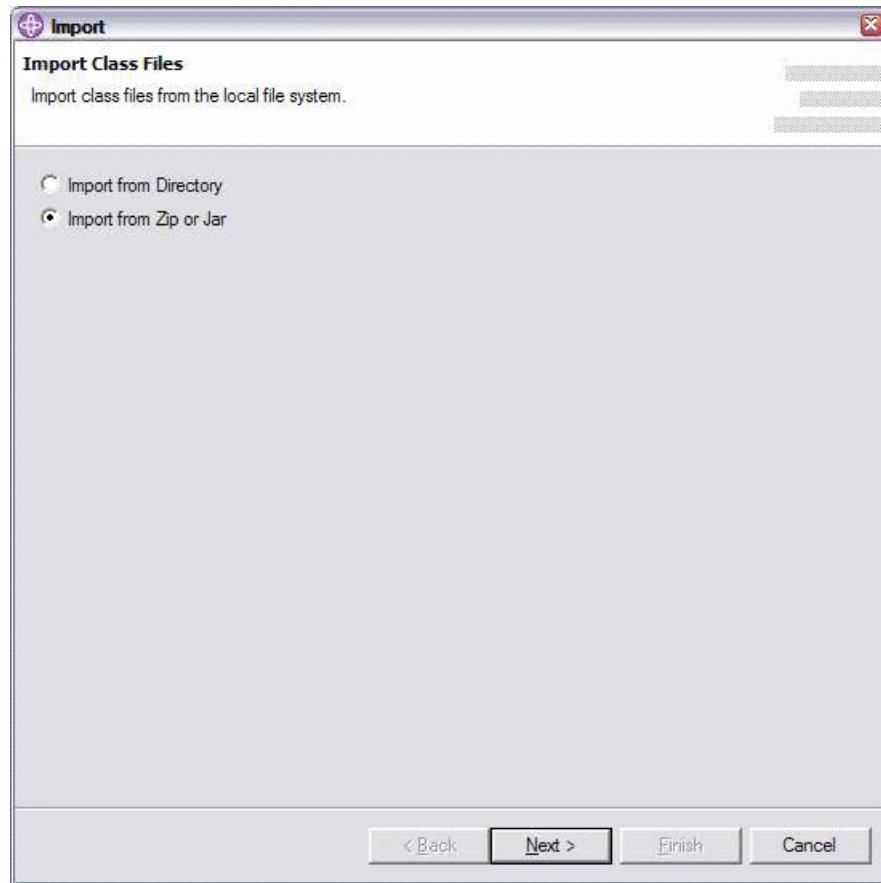


Figure 8-19 Import Class Files panel

14. In the next panel (Figure 8-20), click the **Browse** button and navigate to the location of the ArsWWWServlet.jar file. Click **Select All** and then click **Finish** to complete the import.

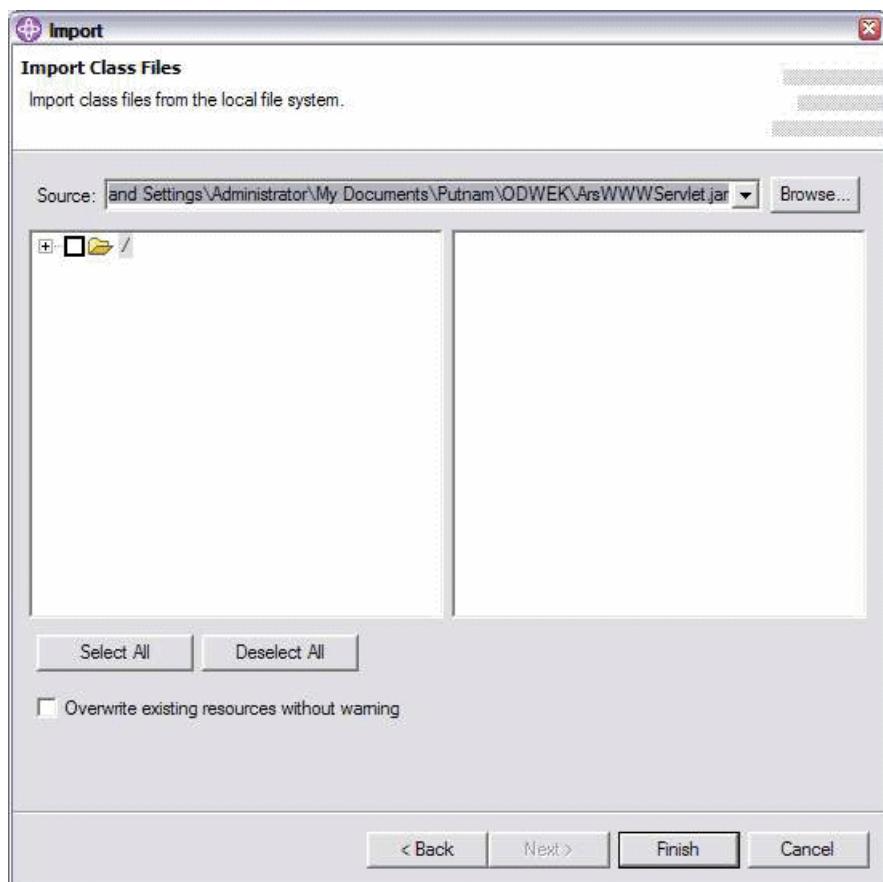


Figure 8-20 Importing class files from the local file system

15. In the J2EE Hierarchy view (Figure 8-21), double-click the **OnDemandWEKWeb** project to open the Web Deployment Descriptor editor.

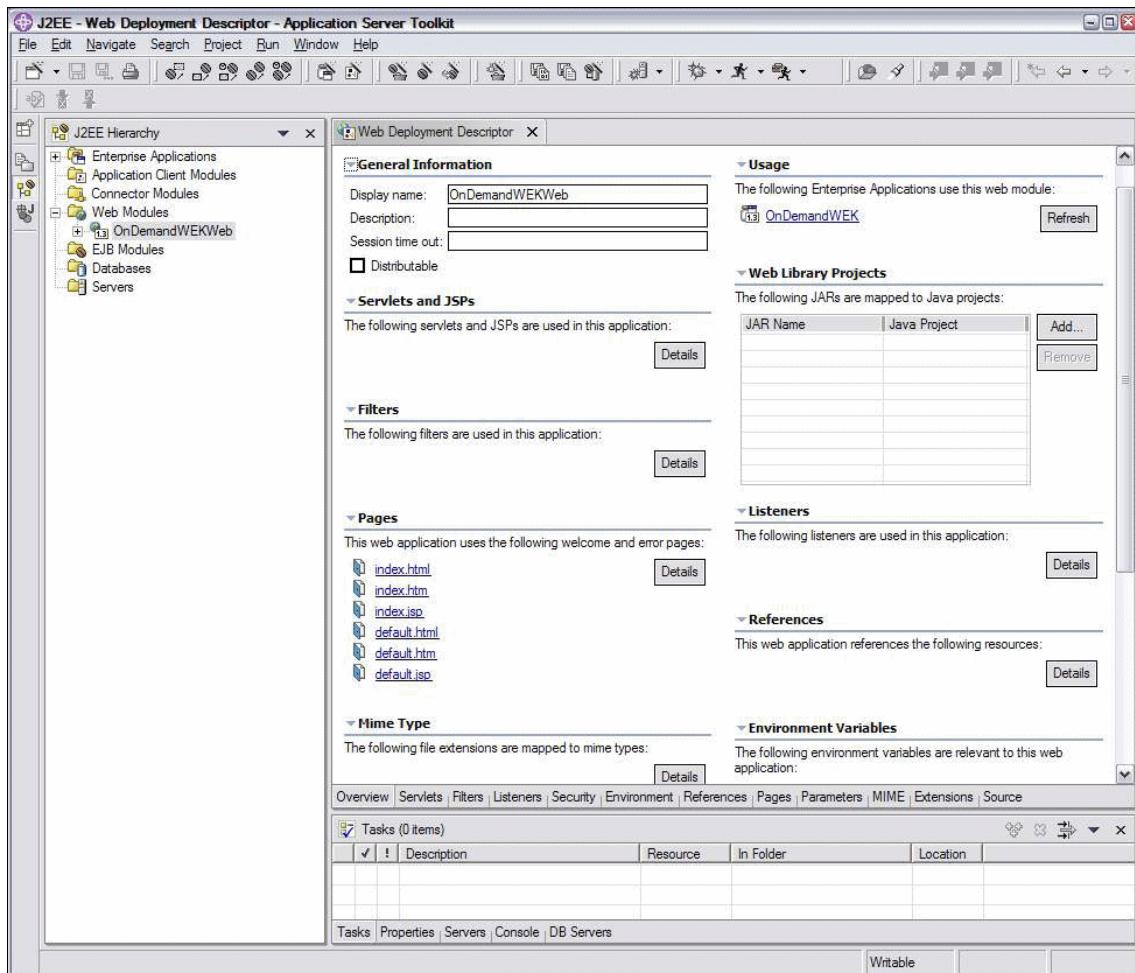


Figure 8-21 Web deployment Descriptor editor

16.in the Web Deployment Descriptor editor (Figure 8-22), click the **Servlets** tab.
Click **Add**.

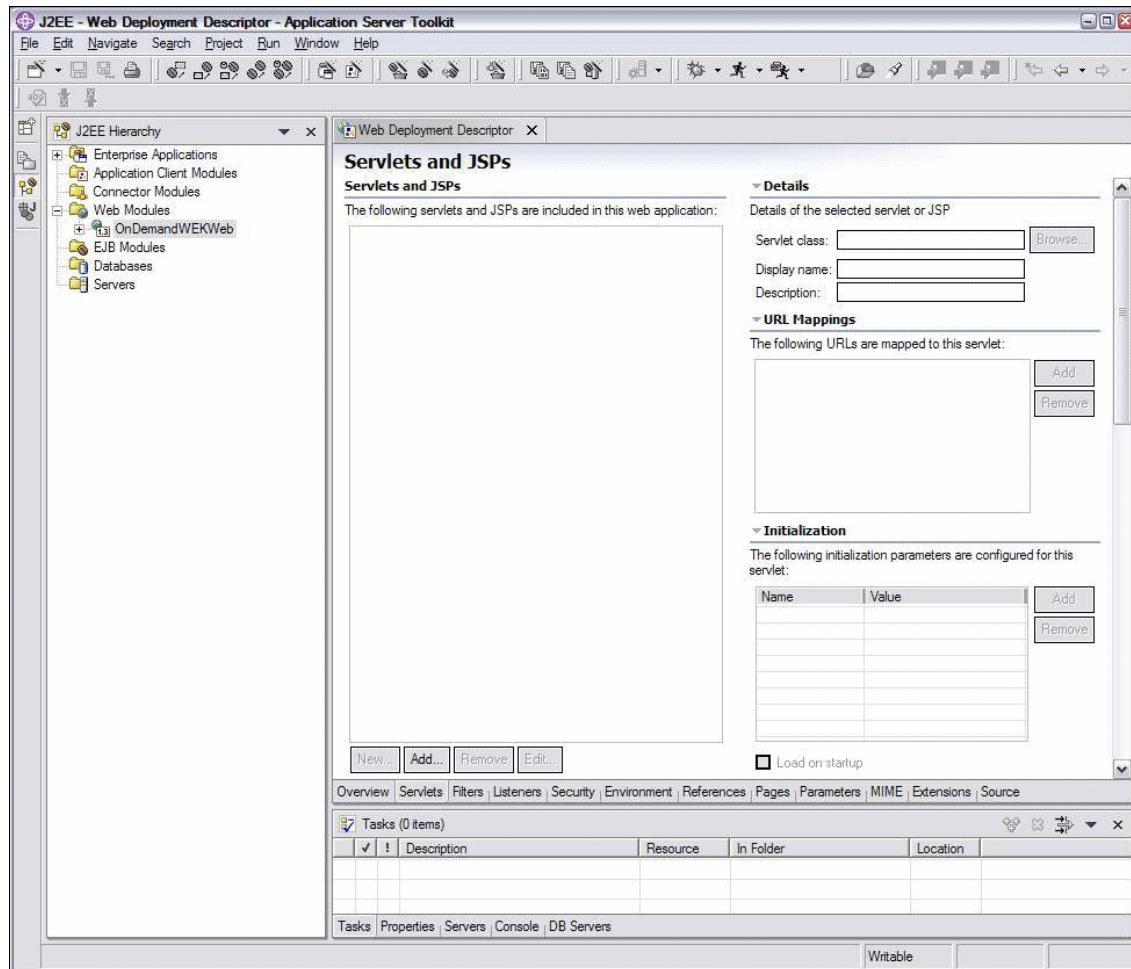


Figure 8-22 Servlets tab

17. In the Add Servlet or JSP window (Figure 8-23), ensure that **Servlet** is selected (the default) and, from the Matching servlets list, choose **ArsWWWServlet**. Then click **OK**.



Figure 8-23 Adding a servlet

18. Add a URL mapping for the ArsWWWServlet. See Figure 8-24. First ensure the **ArsWWWServlet** is selected in the list of servlets. Then in the URL Mappings section, click the **Add** button. This adds a default URL mapping of /ArsWWWServlet.

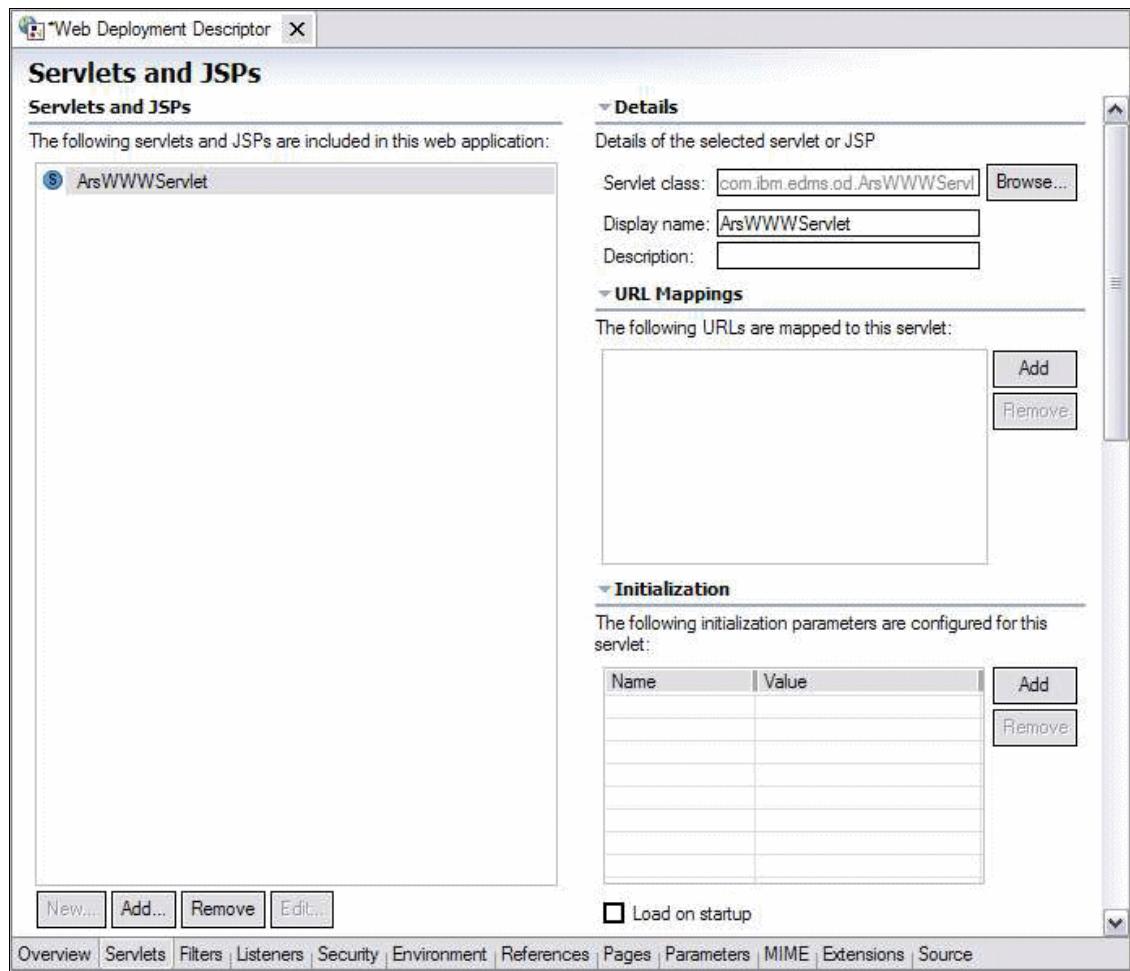


Figure 8-24 Adding a servlet

19. Click the default **ArsWWWServlet** URL Mapping to select it. Enter your desired URL Mapping. See Figure 8-25.

Here you must specify the URL mapping that you want to use when calling the servlet from within the Web browser. The URL mapping includes a user-defined name, for example, od.

Note: You must specify the URL pattern in the format /od/*, where od is the user-defined name.

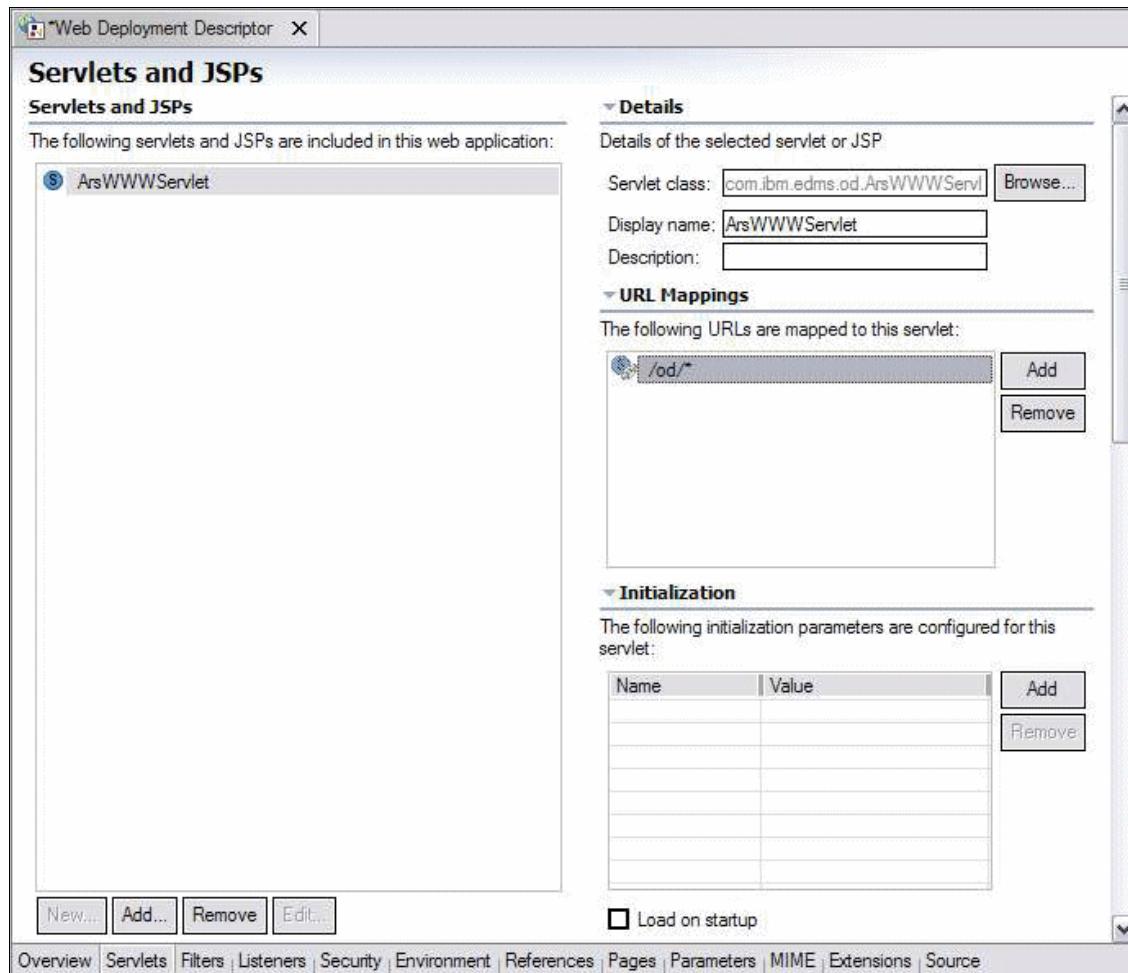


Figure 8-25 URL mapping

20.In the Initialization section, click **Add** to create a new initialization parameter for the ArsWWWServlet servlet. See Figure 8-26. This creates a new parameter with a default name and value.

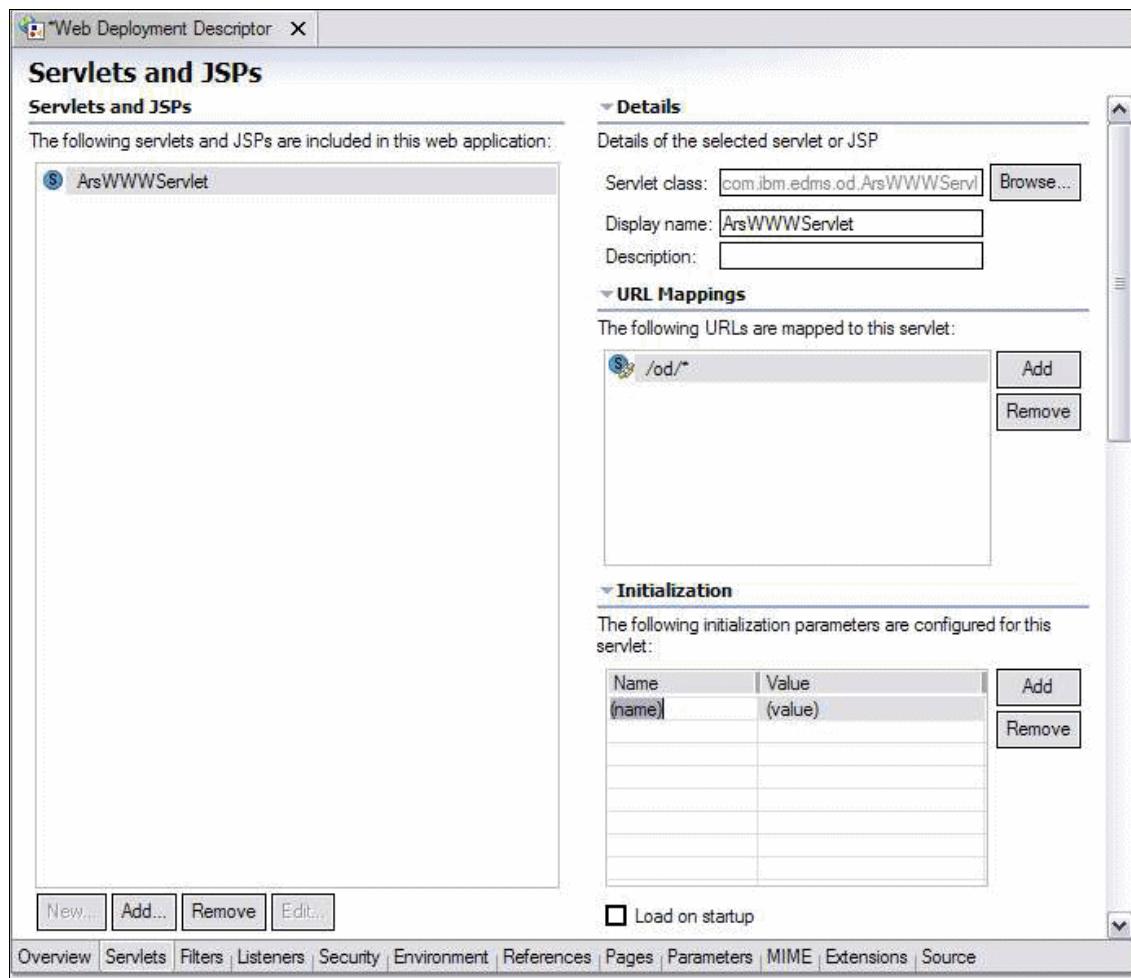


Figure 8-26 Initialization section

21.Select the default name and change it to ConfigDir.

Important: Be sure to use proper capitalization.

22..Select the ConfigDir's parameter value. See Figure 8-27 on page 245.
Change it from the default to one that is appropriate for the runtime platform.
See Table 8-4 on page 245.

Table 8-4 ConfigDir parameter value according to the operating system

Operating system	Value
AIX	/usr/lpp/IBM HTTP Server/bin
HP-UX	/opt/IBM HTTP Server/bin
Linux	/opt/IBM HTTP Server/bin
Solaris	/opt/IBM HTTP Server/bin
Windows	C:\IBM HTTP Server\bin

This value specifies the location of the ARSWWW.INI file. See step 4 on page 227.

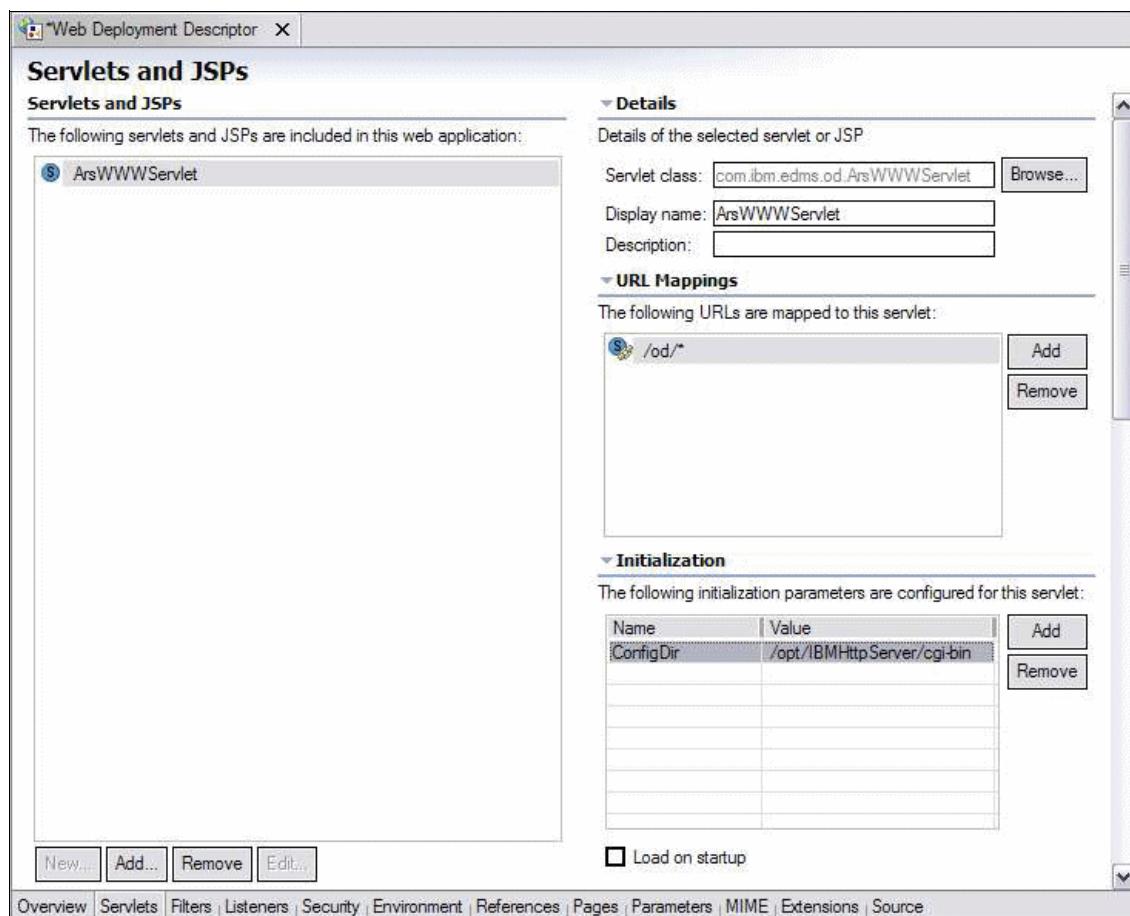


Figure 8-27 ConfigDir parameter value

- 23.Close the Web Deployment Descriptor editor by clicking the **X** next to the view title.
- 24.When prompted to save the deployment descriptor (Figure 8-28), click **Yes**.



Figure 8-28 Web Deployment Descriptor save changes confirmation message

- 25.In the J2EE Hierarchy view, right-click the **OnDemandWEK** enterprise application and select **Export → Export EAR File...**

26. In the EAR Export window (Figure 8-29), specify an EAR file destination (path and file name). Ensure that the option **Export source files** is not selected. Click **Finish**. The EAR file is exported to the destination that you specified in the previous step.

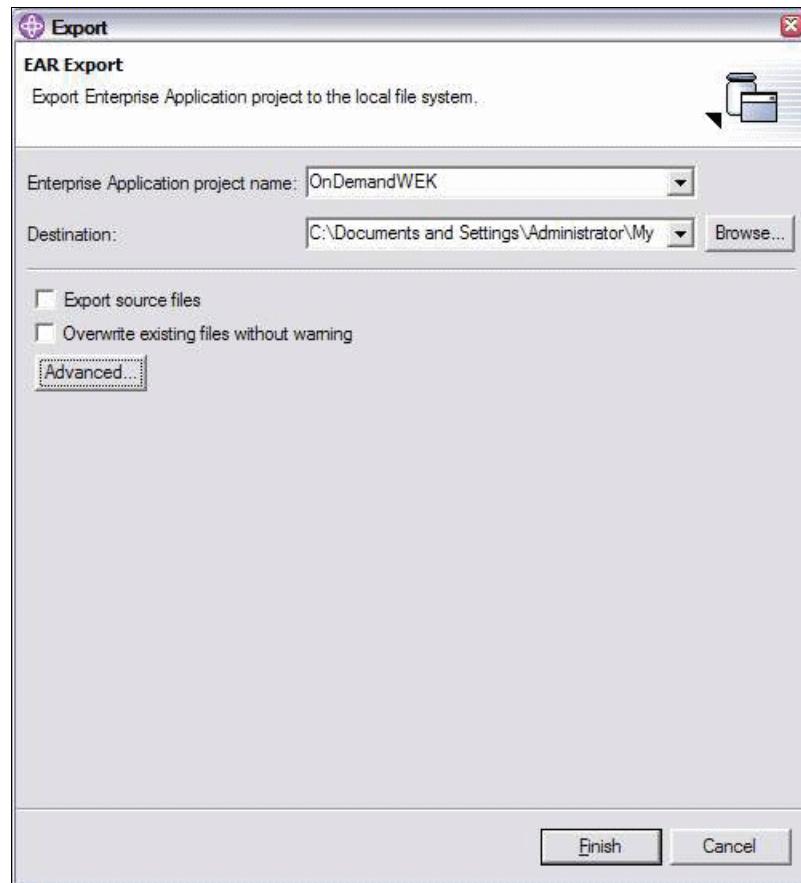


Figure 8-29 EAR Export display

If you follow these steps completely, when you deploy the Web application, the context-root becomes the Web project. When testing the URL to access the servlet, you enter:

`http://hostname/OnDemandWEKWeb/od/arswww`



Data conversion

In this chapter, we provide information about data conversion. We discuss the reasons for data conversion and describe different ways to convert data. We mainly focus on Advanced Function Presentation (AFP), Portable Document Format (PDF), Hypertext Markup Language (HTML), and Extensible Markup Language (XML) flows. AFP is the leading high volume printing data stream, PDF is a frequently required presentation data stream, and HTML and even more so XML are emerging technologies.

We describe two conversion solutions that integrate with OnDemand: the IBM *AFP2WEB Services Offerings* from the team who created AFP data stream and *Xenos d2e from Xenos*. We also explain how to index composed AFP documents that are generated without the requisite tags for indexing.

In this chapter, we cover the following topics:

- ▶ Overview of data conversion
- ▶ IBM AFP2WEB Services Offerings and OnDemand
- ▶ Xenos and OnDemand

9.1 Overview of data conversion

To work with data conversion, it is important that you understand which data conversions are required, and when and how to convert the data. Perform detailed planning before you begin building your solution to help you to achieve a design that remains efficient for many years to come.

In this section, we discuss why to perform data conversion, when to perform it, and how to make the conversion.

9.1.1 Why convert data streams

There are many reasons why you may want to convert data streams. Some of the reasons are as follows:

- ▶ Some data streams, such as Hewlett-Packard (HP) Printer Command Language (PCL) or Xerox metacode, are printer specific and are not displayable. Before archiving or displaying the documents, these data streams must be transformed into a compatible format.
- ▶ The archived data stream might have to comply with company's internal rules or regulations. Hence produced data streams must be transformed into the defined required final flow before being archived.
- ▶ The documents might need to be accessible by people outside the company. The flow should be displayable through standard tools 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 them are displayed in a personalized way. XML flow is an emerging one that allows adaptations and standard exchanges.

9.1.2 When to convert data streams

The decision of *when* to convert data streams relies mainly on the usage 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 on a daily basis. It might also depend on legal requirements about the format of stored data.

We briefly discuss two different data types and the factors that effect this decision:

- ▶ Xerox metacode
- ▶ AFP to PDF

Xerox metacode

Xerox metacode is designed in such a way that all the presentation resources are stored on the actual printer. Because the resources exist on the actual printer and not on the computer, the Xerox metacode cannot be displayed on any computer. Xerox metacode can only be reprinted to a printer that contains the original resources for the document.

If you want to display Xerox metacode through an OnDemand client, you must convert it to something else. If you intend to use a PC client, you must convert the metacode to AFP or PDF at load time since the thick client does not support retrieval transform. If you intend to use OnDemand to reprint the metacode documents to the original metacode printer, then the documents must be stored as metacode. When the documents are stored as metacode, the only way to view them is to enable the retrieval transform through OnDemand Web Enablement Kit (ODWEK) and convert them to either HTML, PDF, or XML.

AFP to PDF

If there is a requirement to present AFP documents in the PDF format over the Web, the best way is to store the documents in their native format and then convert them to PDF at retrieval time. This is because AFP documents are stored much more efficiently than PDF. When multiple AFP documents refer to the same resources, these resources are stored only one time and are shared among the AFP documents.

PDF documents are completely opposite. All the resources necessary to present a PDF document are contained within the document. The PDF document is larger than the original AFP, and the entire print stream, when it is divided into separate customer statements, is much larger, because each individual statement holds all the resources. See 7.2, “Indexing issues with PDF” on page 188, for an example of how a 100k PDF document can be indexed as five separate PDF documents, with a total of 860k in the resulting indexed file size.

Timing is essential to the decision as well. The amount of time needed to convert the document depends on how large it is and how many resources or fonts are associated with the document.

9.1.3 How to convert the data: integrated solutions with OnDemand

Two data conversion solutions can be integrated with OnDemand:

- ▶ IBM AFP2WEB Services Offerings
- ▶ Xenos d2e

IBM AFP2WEB solution is a service offering by the team who created AFP, and it focuses mainly on AFP.

Xenos d2e offers the complementary services in addition to AFP such as HP PCL and Xerox metacode support.

Here are some considerations for target flows:

- ▶ HTML might be used with the same intent, but an HTML flow is not always displayed identically depending to the Web browser used. Additional testing that accounts for the 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 free tools such as Adobe Acrobat Reader. The transformed documents should be displayable, savable, and printable the same way regardless of the environment on which the user is working.
- ▶ XML is an intermediate text-based data stream that allows for the manipulation of documents, regardless of the source data stream, and displays them totally or partially in a personalized way. The use of XML usually involves additional developments including scripts and style sheets.

In the following two sections, we discuss the supported environments, the main functions, and the way these solutions integrate with OnDemand. We also provide some samples.

9.2 IBM AFP2WEB Services Offerings and OnDemand

AFP2PDF and AFP2HTML work in a similar manner. The main difference between these two solutions, beside the different target flows are concerned, is that AFP2PDF takes in account the security options offered by the PDF flow. It might be an additional element of choice between PDF and HTML.

Figure 9-1 shows the AFP2PDF and AFP2HTML transform process.

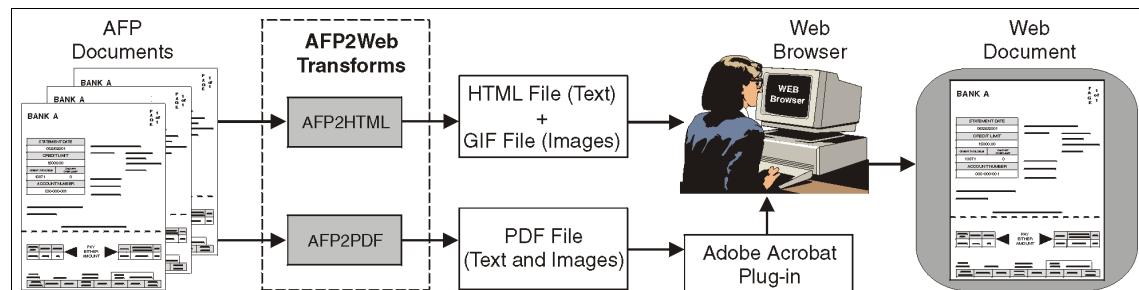


Figure 9-1 AFP2PDF and AFP2HTML

AFP2PDF and AFP2HTML are mutually exclusive. They are both available on Windows NT and 2000, AIX, Sun Solaris, Linux, HP-UX, and z/OS versions. The AIX version also operates in the Qshell environment of the iSeries server. AFP2PDF and AFP2HTML are available on all ODWEK-supported platforms.

9.2.1 AFP2HTML: converting AFP to HTML

AFP2HTML converts your AFP documents to HTML text files and GIF image files. You can display the resulting output with a Web browser, such as Firefox or Microsoft Internet Explorer®. By default, the documents are displayed with the AFP2HTML applet. However, because the AFP2HTML conversion cannot exactly map AFP format to HTML format, certain limitations and approximations exist when you view the output with the Web browser.

How AFP2HTML works with ODWEK

AFP2HTML works with ODWEK through the configurations of the following parameters and files:

- ▶ The AFPVIEWING parameter in the arswww.ini file
- ▶ The [AFP2HTML] section in the arswww.ini file
- ▶ The configuration file, afp2html.ini, specified by the CONFIGFILE parameter in the arswww.ini file
- ▶ Control files

The AFPVIEWING parameter in arswww.ini file

When a user retrieves an AFP document from the OnDemand server, the value of the AFPVIEWING parameter in the arswww.ini file determines what action, if any, ODWEK takes before sending the document to the viewer.

AFPVIEWING might have the following values:

- ▶ PLUGIN
This indicates that ODWEK does not convert AFP documents (which is the default) and the plug-in has to be installed on the users' computers.
- ▶ HTML
This indicates that ODWEK converts AFP documents to HTML documents with the AFP2HTML. Additional information can be found in the [AFP2HTML] section of the arswww.ini file.
- ▶ PDF
This indicates that ODWEK converts AFP documents to PDF documents with AFP2PDF. Additional information can be found in the [AFP2PDF] section of the arswww.ini file.

Example 9-1 shows the sample lines in arswww.ini file, where AFP is set to convert to HTML.

Example 9-1 The arswww.ini file: AFP converted to HTML

```
[DEFAULT BROWSER]  
AFPVIEWING=HTML
```

The [AFP2HTML] section in the arswww.ini file

The [AFP2HTML] section contains the parameters that are used by AFP2HTML. These parameters are:

► **CONFIGFILE**

This parameter specifies the configuration file that contains the options used by AFP2HTML to convert APF documents and resources into HTML data, fonts, and images.

► **INSTALLDIR**

This parameter specifies the directory that contains AFP2HTML programs, configurations files, and mapping files.

► **USEEXECUTABLE**

This setup determines whether ODWEK starts AFP2HTML by using the shared library or the executable. The default value is 0 and means that ODWEK uses the shared library. This value should assure better performance. When the transform is used in a Java environment, USEEXECUTABLE=1 might be necessary if memory issues are submitted from the Java virtual machine (JVM™).

Example 9-2 shows the sample lines of the [AFP2HTML] section in an arswww.ini file, where AFP is set to convert to HTML.

Example 9-2 The arswww.ini file: AFP2HTML section

```
[AFP2HTML]  
CONFIGFILE=afp2html.ini  
INSTALLDIR=/usr/lpp/ars/www/bin/afp2html  
USEEXECUTABLE=0
```

The configuration file, *afp2html.ini*, specified in the *arswww.ini* file

Example 9-3 provides a sample configuration file, *afp2html.ini*, specified by the *CONFIGFILE* parameter in the *arswww.ini* file.

Example 9-3 The afp2html.ini file

```
[CREDIT-CREDIT]
UseApplet=TRUE
ScaleFactor=1.0
CreateGIF=TRUE
FontMapFile=creditFontMap.cfg
ImageMapFile=creditImageMap.cfg

[default]
ScaleFactor=1.0
CreateGIF=TRUE
FontMapFile=fontmap.cfg
ImageMapFile=imagemap.cfg
```

The structure of the *afp2html.ini* file is similar to a Windows INI file. It contains one section for each AFP application and a default section. The title line of the section identifies the application group and application.

For example, the title line [CREDIT-CREDIT] identifies the CREDIT application group and the CREDIT application. Use the – (dash) character to separate the names in the title line. The names must match the application group and application names defined to the OnDemand server. If the application group contains more than one application, then create one section for each application.

The options in the [default] section are used by AFP2HTML to process documents for AFP applications that are not identified in the AFP2HTML.INI file. The defaults are also used if an AFP application section does not include one of the options.

The *UseApplet* option is a directive to ODWEK. It determines whether the AFP2HTML applet will be used to view the output from the AFP2WEB transform. The default value is TRUE. If you specify FALSE (the AFP2HTML applet is not used to view the output), the output is formatted and displayed by the Web browser.

Note: The Java AFP2HTML Viewer is an applet that enables users to view the output generated by AFP2HTML. AFP2HTML converts AFP documents and resources into HTML documents. The Java AFP2HTML Viewer provides a toolbar with controls that can help users to work with documents, such as pagination and annotation, including controls for large objects.

One advantage of the applets is that users never have to install or upgrade software on the PC to use them. When using the applets and viewers that are provided by IBM, the documents that are retrieved from an OnDemand server remain compressed until they reached the viewer.

The viewer uncompresses the documents and displays the pages in a Web browser window. If a document is stored in OnDemand as a large object, the viewer retrieves and uncompresses segments of the document, as needed, when the user moves through the pages of the document.

The *AllObjects* parameter determines how ODWEK processes documents that are stored as large objects in OnDemand. The default value is 0, and it means that ODWEK retrieves only the first segment of a document. If you specify a value of 1, then ODWEK retrieves all of the segments and converts them before sending the document to the viewer.

Note: If you enable Large Object support for very large documents and specify 1, then your users might experience a significant delay before they can view the document.

The *ScaleFactor* parameter scales the output with the given scale factor. The default value is 1.0. For example, specifying a value of ScaleFactor=2.0 scales the output to be twice as large as the default size; specifying a value of ScaleFactor=0.5 scales the output to one half of the default size. The default size is derived from the Zoom setting on the Logical Views page in the OnDemand application.

The *SuppressFonts* parameter determines whether the AFP text strings are transformed. If you specify SuppressFonts=TRUE, any text that uses a font listed in the Font Map file is not transformed. The default value is FALSE, which means that all of the AFP text strings are transformed. The Font Map file is identified with the FontMapFile option.

The *FontMapFile* parameter identifies the full path name of the Font Map file. The Font Map file contains a list of fonts that require special processing. The default Font Map file is named `imagfont.cfg` and resides in the directory that contains the AFP2HTML programs. See the AFP2WEB transform documentation for details about the Font Map file.

The *ImageMapFile* parameter identifies the image mapping file. The image mapping file can be used to remove images from the output, improve the look of shaded images, and substitute existing images for images created by the AFP2HTML transform. Mapping images that are common across your AFP documents (for example, a company logo) reduces the time required to transform documents. If specified, the image mapping file must exist in the directory that contains the AFP2HTML programs. See the AFP2WEB transform documentation for details about the image mapping file.

Control files

Three control files are used by AFP2HTML:

- ▶ The *font map file*, by default `imagfont.cfg`, is listed in the `afp2html.ini` file. See the AFP2WEB transform documentation for details about this file.
- ▶ The *image map file* is listed in the `afp2html.ini` file. We present some samples about the way to use it in “Mapping AFP images” on page 258.
- ▶ The *transform profile file* by default, is `afp2web.ini`, with parameters to control settings for AFP2HTML:
 - *ResourceDataPath* specifies the directories that the transform uses to search for AFP resources.
 - *FontDataPath* specifies the base directory that the transform uses to search for the font configuration files.
 - *PageSegExt* sets the file extension to be used when searching for a page segment file in a resource directory. For example, if all the page segment resource files had the file extension of `.PSG`, you can set it as:
`PageSegExt=*.PSG`
 - *OverlayExt* sets the file extension to be used when searching for an overlay file in a resource directory. For example, if all of the overlay resource files had the file extension of `.OLY`, you can set it as:
`OverlayExt=*.OLY`

Mapping AFP images

When an AFP document is transformed, images are identified using parameters that specify the page segment name (if available), the position, and the size of each image. If an AFP page contains images, AFP2HTML creates image information entries as comments in the HTML file (AFP2HTML). The HTML comments can be copied into an image map configuration file to define and map a particular image.

Mapping images gives you the option of handling AFP images in different ways during the transform process, including:

- ▶ Removing images

You can remove all or some of the images from your transformed output.

- ▶ Substituting existing images

You can substitute all or some of the images with previously generated Internet images in the HTML output, such as Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG), or other images.

- ▶ Substituting AFP shaded images with colored areas

You can substitute all or some of the images with a solid-colored rectangle. This is especially useful for improving the look of the shaded areas that are defined as images in the AFP data stream.

- ▶ Adding an image that is not in the AFP data

You can add an image, which is not part of the AFP data, to the HTML or PDF display that models the use of a preprinted form used during printing.

The configuration file handles all the transform processing for the images. For example, when the transform program is run against an AFP document and an image is encountered, the program looks for a matching image entry in the configuration file. If an entry is defined that matches the name, position, size, or a combination of the three, the information in the configuration file is used for the transform of the image.

Creating the image map file

To map images for your AFP files, you must create an image map configuration file. The best way to do this is to transform a sample AFP document that represents all documents that will use the configuration file. You then identify the image entries that are created during the transform and define them in the configuration file.

Run the **afp2web** command using the afpdoc AFP document:

```
afp2web c:\documents\afpdoc.afp
```

We get the c:\documents\afpdoc.html file, which contains HTML comments for each image as shown in Example 9-4.

Example 9-4 Image map file

```
<!-- IMAGE position:(5.250in,0.613in) / (567px,66px) size:(0.667in,0.800in /
(72px,86px) -->
<!-- IMAGE_END -->
<!-- IMAGE position:(0.863in,8.483in) / (93px,916px) size:(2.400in,0.667in) /
(259px,72px) -->
<!-- IMAGE_END -->
<!-- IMAGE position:(3.596in,8.550in) / (388px,923px) size:(2.633in,0.700in) /
(284px,76px) -->
<!-- IMAGE_END -->
<!-- IMAGE name:(S1PSEG01) position:(6.162in,8.483in) / (666px,916px)
size:(2.067in,0.604in) / (223px,65px) -->
<!-- IMAGE_END -->
```

Now it is possible to remove, substitute, or add images.

Removing images

If no other information is given as part of an entry in the image map configuration file, such as extra lines between the image position and size definitions and the IMAGE_END tag, the entry is considered empty. Then the image is removed from the transformed GIF files. The image information that the transform program generates is empty by default.

Substituting existing images

To use an existing image, add IMAGE definition parameters between the starting <!-- IMAGE ...--> and ending <!-- IMAGE_END --> lines of an image information entry in the configuration file. For example, with Example 9-5, when the first image is encountered, it is substituted with logo1.gif, and when the second image is encountered, it is substituted with logo2.gif.

Example 9-5 Substituting images

```
<!-- IMAGE position:(5.250in,0.613in) / (567px,66px) size:(0.667in,0.800in /
(72px,86px) -->
IMAGE XPos=0 YPos=0 XSize=900 YSize=200
URL="http://www.comp_a.com/images/logo1.gif" ZIndex=1
<!-- IMAGE_END -->
<!-- IMAGE position:(0.863in,8.483in) / (93px,916px) size:(2.400in,0.667in) /
(259px,72px) -->
IMAGE XPos=0 YPos=0 XSize=500 YSize=300 URL=images/logo2.gif ZIndex=2
<!-- IMAGE_END -->
<!-- IMAGE position:(3.596in,8.550in) / (388px,923px) size:(2.633in,0.700in) /
(284px,76px) -->
<!-- IMAGE_END -->
```

```
<!-- IMAGE name:(S1PSEG01) position:(6.162in,8.483in) / (666px,916px)  
size:(2.067in,0.604in) / (223px,65px) -->  
<!-- IMAGE_END -->
```

Substituting AFP shaded images with colored areas

Many AFP documents contain areas on the page that are shaded with a gray box. This is accomplished in the AFP data stream by defining an image that has pixels laid out in a regular checker-board-like pattern to create a gray shading effect. When attempting to display this type of image, however, it often becomes distorted due to the scale factor and resolution of the display hardware. To avoid this problem, you can define a colored area to use instead of the shaded image.

To substitute a shaded image with a color area, add COLORED_AREA definition parameters between the starting <!-- IMAGE ... --> and ending <!-- IMAGE_END --> lines of an image information entry in the configuration file. For example with Example 9-6, when the first image is encountered, it is substituted with a red, 86x72 pixel area positioned at (567,66).

Example 9-6 Substituting shaded area

```
<!-- IMAGE position:(5.250in,0.613in) / (567px,66px) size:(0.667in,0.800in) /  
(72px,86px) -->  
COLORED_AREA XPos=567 YPos=66 XSize=72 YSize=86 Color=red  
<!-- IMAGE_END -->  
<!-- IMAGE position:(0.863in,8.483in) / (93px,916px) size:(2.400in,0.667in) /  
(259px,72px) -->  
<!-- IMAGE_END -->  
<!-- IMAGE position:(3.596in,8.550in) / (388px,923px) size:(2.633in,0.700in) /  
(284px,76px) -->  
<!-- IMAGE_END -->  
<!-- IMAGE name:(S1PSEG01) position:(6.162in,8.483in) / (666px,916px)  
size:(2.067in,0.604in) / (223px,65px) -->  
<!-- IMAGE_END -->
```

Adding an image that is not in the AFP data

Occasionally, preprinted forms are used during the printing process. These preprinted forms might have a company logo, a table, or grid that is filled in with the print data. The transforms enable an image, which emulates the preprinted form, to be included in the transformed output. This image can be in color and can be included on all the pages or only the first page created.

To include an image that emulates a preprinted form, add the STATIC_IMG definition parameters between the <!-- IMAGE --> and <!-- IMAGE_END --> lines in the configuration file. These starting and ending lines are special in that they do not include any name, position, or size information. You must manually add these starting and ending lines as well as the STATIC_IMG definition to the configuration file. For example, as shown in Example 9-7, the form1.gif GIF image is included on all pages in the HTML output.

Example 9-7 Emulating a preprinted form

```
<!-- IMAGE -->
STATIC_IMG XPos=0 YPos=0 XSize=72 YSize=722
URL=="http://www.comp_a.com/images/logo1.gif" Type=ALL ZIndex=1
<!-- IMAGE_END -->
<!-- IMAGE position:(0.863in,8.483in) / (93px,916px) size:(2.400in,0.667in) /
(259px,72px) -->
<!-- IMAGE_END -->
<!-- IMAGE position:(3.596in,8.550in) / (388px,923px) size:(2.633in,0.700in) /
(284px,76px) -->
<!-- IMAGE_END -->
<!-- IMAGE name:(S1PSEG01) position:(6.162in,8.483in) / (666px,916px)
size:(2.067in,0.604in) / (223px,65px) -->
<!-- IMAGE_END -->
```

AFP2HTML command

As seen before, the conversion function is called automatically from OnDemand. However for different purposes such as creating the Image Map File or testing the conversion result, the **afp2web** command might be used.

See the AFP2WEB transform documentation for details about the **afp2web** command parameters.

9.2.2 AFP2PDF: converting AFP to PDF

AFP2PDF converts your AFP documents to the Adobe PDF format so you can view them with Adobe Acrobat. If the Adobe Acrobat plug-in is installed with the Web browser, you can view and print these documents within the browser application. AFP2PDF exactly maps AFP format to PDF format, making it a more robust solution than AFP2HTML. However, to display this data, the Adobe Acrobat software must be installed on the client workstation.

How AFP2PDF works with ODWEK

AFP2PDF works with ODWEK through the configurations of the following parameters and files:

- ▶ The AFPVIEWING parameter in the arswww.ini file
- ▶ The [AFP2PDF] section in the arswww.ini file
- ▶ The configuration file, afp2pdf.ini, specified by the CONFIGFILE parameter in the arswww.ini file
- ▶ Control files

The AFPVIEWING parameter in arswww.ini file

When a user retrieves an AFP document from the OnDemand server, the value of the AFPVIEWING parameter in the arswww.ini file determines what action, if any, ODWEK takes before sending the document to the viewer.

- ▶ PLUGIN

This value indicates that ODWEK does not convert AFP documents (the default) and the plug-in must be installed on the users' computers.

- ▶ HTML

This value indicates that ODWEK converts AFP documents to HTML documents with the AFP2HTML. You can find additional information in the [AFP2HTML] section of the arswww.ini file.

- ▶ PDF

This means that ODWEK converts AFP documents to PDF documents with AFP2PDF. You can find additional information in the [AFP2PDF] section of the arswww.ini file.

Example 9-8 shows the sample lines in the arswww.ini file, where AFP is set to convert to PDF.

Example 9-8 The arswww.ini file: AFP converted to PDF

```
[DEFAULT BROWSER]
AFPVIEWING=PDF
```

The [AFP2PDF] section in the arswww.ini file

The [AFP2PDF] section contains the parameters that are used by either AFP2PDF. These parameters are:

- ▶ CONFIGFILE

This parameter specifies the configuration file that contains the options used by AFP2PDF to convert AFP documents and resources into PDF documents.

► **INSTALLDIR**

This parameter specifies the directory that contains AFP2PDF programs, configurations files, and mapping files.

► **USEEXECUTABLE**

This parameter determines whether ODWEK starts AFP2PDF by using the shared library or the executable. The default value is 0, and it means that ODWEK uses the shared library. This value should assure better performance. When the transform is used in a Java environment, USEEXECUTABLE=1 might be necessary if memory issues are submitted from JVM.

Example 9-9 shows the sample lines of the [AFP2PDF] section in the arswww.ini file, where AFP is set to convert to PDF.

Example 9-9 The arswww.ini file: AFP2PDF section

```
[AFP2PDF]
CONFIGFILE=afp2pdf.ini
INSTALLDIR=/usr/lpp/ars/www/bin/afp2pdf
USEEXECUTABLE=0
```

The configuration file, afp2pdf.ini, specified in the arswww.ini file

Example 9-10 provides a sample configuration file, afp2pdf.ini, specified by the CONFIGFILE parameter in the arswww.ini file.

Example 9-10 The afp2pdf.ini file

```
[CREDIT-CREDIT]
OptionsFile=
ImageMapFile=creditImageMap.cfg

[default]
OptionsFile=
ImageMapFile=imagemap.cfg
AllObjects=0
```

The structure of the file is similar to a Windows INI file. It contains one section for each AFP application and a default section. The title line of the section identifies the application group and application.

For example, the title line, [CREDIT-CREDIT] identifies the CREDIT application group and the CREDIT application. Use the – (dash) character to separate the names in the title line. The names must match the application group and the application names defined to the OnDemand server. If the application group contains more than one application, then create one section for each application.

The parameters that you specify in the [default] section are used by the AFP2PDF transform to process documents for AFP applications that are not identified in the `afp2pdf.ini` file. The default parameters are also used if an AFP application section does not include one of the specified parameters.

The *OptionsFile* parameter identifies the full path name of the file that contains the transform options used by the AFP2PDF transform. The transform options are used for AFP documents that require special processing. The *ImageMapFile* parameter identifies the image mapping file.

The image mapping file can be used to remove images from the output, improve the look of shaded images, and substitute existing images for images created by the AFP2PDF transform. Mapping images that are common in most of your AFP documents (such as a company logo) reduce the time required to transform documents. If specified, the image mapping file must exist in the directory that contains the AFP2PDF transform programs.

The *AllObjects* parameter determines how ODWEK processes documents that are stored as large objects in OnDemand. The default value is 0, and it means that ODWEK retrieves only the first segment of a document. If you specify a value of 1, then ODWEK retrieves all of the segments and converts them before sending the document to the viewer.

Note: If you enable Large Object support for very large documents and specify a value of 1, then your users might experience a significant delay before they can view the document.

Control files

Two control files, listed in the `afp2pdf.ini` file, are used by AFP2PDF:

- ▶ The *image map file*: We present some samples for using it in “Mapping AFP images” on page 264.
- ▶ The *options file*, by default `a2pxopts.ini`, with parameters to control settings for AFP2HTML: We present some of the most important parameters in “Option file” on page 269.

Mapping AFP images

When an AFP document is transformed, images are identified using parameters that specify the page segment name (if available), the position, and the size of each image. If an AFP page contains images, AFP2PDF creates image information entries in an output file (AFP2PDF). The output file entries can be copied into an image map configuration file to define and map a particular image.

Mapping images gives you the option of handling AFP images in different ways during the transform process, including:

- ▶ Removing images
 - You can remove all or some of the images from your transformed output.
- ▶ Substituting existing images
 - You can substitute all or some of the images with previously generated images in the PDF output with JPEG images.
- ▶ Substituting AFP shaded images with colored areas
 - You can substitute all or some of the images with a solid-colored rectangle. This is especially useful for improving the look of the shaded areas that are defined as images in the AFP data stream.
- ▶ Adding an image not in the AFP data
 - You can add an image that is not part of the AFP data to the PDF display that models the use of a preprinted form used during printing.
- ▶ Caching frequently used images
 - You can cache frequently used images to reduce the size of a PDF file.

The configuration file handles all the transform processing for the images. For example, when the transform program is run against an AFP document and an image is encountered, the program looks for a matching image entry in the configuration file. If an entry is defined that matches the name, position, size, or a combination of the three, the information in the configuration file is used for the transform of the image.

Creating the image map file

To map images for your AFP files, you must create an image map configuration file. The best way to do this is to transform a sample AFP document that represents all documents that will use the configuration file. You then identify the image entries created during the transform and define them in the configuration file.

Run the **afp2pdf** command using the afpdoc AFP document:

```
afp2pdf c:\documents\afpdoc.afp
```

We get two files:

- ▶ c:\documents\afpdoc.pdf (PDF file for the AFP document)
- ▶ c:\imagemap.out (a file with image information for the AFP document; see Example 9-11)

Note: The image information file is generated according the ImageMapEntries_File parameter in AFP2PDF Options File. Refer to “Control files” on page 264.

Example 9-11 Image map file

```
<IMAGE position:(5.250in,0.613in) size:(0.667in,0.800in)>
<IMAGE_END>
<IMAGE position:(0.863in,8.483in) size:(2.400in,0.667in)>
<IMAGE_END>
<IMAGE position:(3.596in,8.550in) size:(2.633in,0.700in)>
<IMAGE_END>
<IMAGE name:(S1PSEG01) position:(6.162in,8.483in) size:(2.067in,0.604in)>
<IMAGE_END>
```

The image information for each image is in pairs. The first line contains the page-segment resource name (only if available), the position value in inches, and size values in inches. The second line ends the entry for the image. The first value for the position and size gives the horizontal dimension and the second gives the vertical dimension. The position measurements are for the upper, left-hand corner of the image relative to the upper, left-hand corner of the page.

The copy of the lines in the output file (imagemap.out in this example) is used to create the image-map configuration file (imagemap.cfg by default).

The image information in the configuration file is used to identify the images in the AFP document during the transform process. Each IMAGE tag along with its corresponding IMAGE_END tag defines a single image information entry in the configuration file.

Now it is possible to remove, substitute, or add images.

Removing images

If no other information is given as part of an entry in the image map configuration file, such as extra lines between the image position and size definitions and the IMAGE_END tag, the entry is considered empty and the image is removed from the transformed PDF files. The image information that the transform program generates is empty by default.

Substituting existing images

To use an existing image, add IMAGE definition parameters between the starting <IMAGE ...> and ending <IMAGE_END> lines of an image information entry in the configuration file. For example, with Example 9-12, when the first image is encountered, it is substituted with logo1.jpg, and when the second image is encountered, it is substituted with logo2.jpg.

Example 9-12 Substituting images

```
<IMAGE position:(5.250in,0.613in) / (567px,66px) size:(0.667in,0.800in) /  
(72px,86px)>  
IMAGE XPos=0 YPos=0 XSize=900 YSize=200 Filename="/images/logo1.jpg"  
<IMAGE_END>  
<IMAGE position:(0.863in,8.483in) / (93px,916px) size:(2.400in,0.667in) /  
(259px,72px)>  
IMAGE XPos=0 YPos=0 XSize=500 YSize=300 Filename="/images/logo2.jpg"  
<IMAGE_END>  
<IMAGE position:(3.596in,8.550in) / (388px,923px) size:(2.633in,0.700in) /  
(284px,76px)>  
<IMAGE_END>  
<IMAGE name:(S1PSEG01) position:(6.162in,8.483in) / (666px,916px)  
size:(2.067in,0.604in) / (223px,65px)>  
<IMAGE_END>
```

Substituting AFP shaded images with colored areas

Many AFP documents contain areas on the page that are shaded with a gray box. This is accomplished in the AFP data stream by defining an image that has pixels laid out in a regular checker-board-like pattern to create a gray shading effect. When attempting to display this type of image, however, it often becomes distorted due to the scale factor and resolution of the display hardware. To avoid this problem, you can define a colored area to be used instead of the shaded image.

To substitute a shaded image with a color area, add SHADED_AREA definition parameters between the starting <IMAGE ...> and ending <IMAGE_END> lines of an image information entry in the configuration file. For example, with Example 9-13, when the first image is encountered, it is substituted with a red, 0.667in x 0.8in area positioned at (5.25in,0.613in).

Example 9-13 Substituting shaded area

```
<IMAGE position:(5.250in,0.613in) size:(0.667in,0.800in)>  
SHADED_AREA XPos=5.250 YPos=0.613 XSize=0.667 YSize=0.800 Shade_R=1.0  
Shade_G=0.0 Shade_B=0.0  
<IMAGE_END>  
<IMAGE position:(0.863in,8.483in) size:(2.400in,0.667in)>  
<IMAGE_END>
```

```
<IMAGE position:(3.596in,8.550in) size:(2.633in,0.700in)>
<IMAGE_END>
<IMAGE position:(6.162in,8.483in) size:(2.067in,0.604in)>
<IMAGE_END>
```

Adding an image not in the AFP data

Occasionally, preprinted forms are used during the printing process. These preprinted forms might have a company logo, a table, or grid that is filled in with the print data. The transforms let an image, which emulates the preprinted form, be included in the transformed output. This image can be in color and can be included on all the pages or only the first page created.

To include an image that emulates a preprinted form, add the STATIC_IMG definition parameters between the <IMAGE ...> and ending <IMAGE_END> lines in the configuration file. These starting and ending lines are special in that they do not include position or size information. You must manually add these starting and ending lines as well as the STATIC_IMG definition to the configuration file. In Example 9-14, the form1.jpg JPEG image is included on all pages in the PDF output.

Example 9-14 Emulating a preprinted form

```
< IMAGE>
STATIC_IMG XPos=0 YPos=0 XSize=72 YSize=722 Filename="c:\images\form1.jpg"
Type=ALL
< IMAGE_END>
<IMAGE position:(5.250in,0.613in) size:(0.667in,0.800in)>
<IMAGE_END>
<IMAGE position:(0.863in,8.483in) size:(2.400in,0.667in)>
<IMAGE_END>
<IMAGE position:(3.596in,8.550in) size:(2.633in,0.700in)>
<IMAGE_END>
<IMAGE position:(6.162in,8.483in) size:(2.067in,0.604in)>
<IMAGE_END>.
```

Caching frequently used images

In many AFP documents, the same image is used many times throughout the document, such as a company logo that appears on each page of a document. It is possible to cache this image so that the image is stored only once and referenced each time it is used. Caching frequently used images reduces the size of the resulting PDF file.

To cache an image, add a CACHE_IMG definition parameter between the starting <IMAGE ...> and ending <IMAGE_END> lines of an image information entry in the configuration file. Example 9-15 shows how to cache frequently used images. When the first image is encountered, it is cached with the name IMG0.

Example 9-15 Caching an image

```
<IMAGE position:(5.250in,0.613in) size:(0.667in,0.800in)>
CACHE_IMG NAME=IMG0
<IMAGE_END>
<IMAGE position:(0.863in,8.483in) size:(2.400in,0.667in)>
<IMAGE_END>
<IMAGE position:(3.596in,8.550in) size:(2.633in,0.700in)>
<IMAGE_END>
<IMAGE position:(6.162in,8.483in) size:(2.067in,0.604in)>
<IMAGE_END>
```

Option file

An option file might be associated to specific OnDemand application group and application or defaulted.

We present part of the options that are specific to the PDF flow. These might concern either information made available to the user through options in the PDF viewer or security that limits the user in the actions that can be taken with the document. See the AFP2PDF transform documentation for details about the option file parameters.

Information associated with a PDF document

Options associated with a PDF document include:

- ▶ Append_PDF_file

This parameter appends the listed PDF file to the generated PDF file. For example, the following line appends the file C:\term.pdf to the beginning of the generated PDF file:

```
Append_PDF_File="C:\\term.pdf",0
```

The following line appends the file c:\term.pdf to the end of the generated PDF file:

```
Append_PDF_File="C:\\term.pdf",1
```

- ▶ Author

This parameter specifies the text for the Author entry in the Info Dictionary of the output PDF file. This information is displayed to the user if the “general document info” function is selected in Adobe Acrobat.

- ▶ Creator

This parameter specifies the text for the Creator entry in the Info Dictionary of the output PDF file. This information is displayed to the user if the “general document info” function is selected in Adobe Acrobat.

- ▶ **Disable_Bookmark_Generation**

If page level indexing information is available in the input AFP document, PDF bookmarks are automatically generated. Setting this option to FALSE, the bookmarks are not created.

- ▶ **Keywords**

This parameter sets the text for the Keywords entry in the Info Dictionary of the output PDF file. This information is displayed to the user if the “general document info” function is selected in Adobe Acrobat.

- ▶ **Show_Outline**

If an AFP document contains index data, the transform converts this index information into PDF outline and bookmark functions. If the output PDF file contains any bookmark information, the outline window is always displayed when viewed with Adobe Acrobat. By setting this parameter value to FALSE, the outline window is not displayed.

- ▶ **Subject**

This parameter sets the text for the Subject entry in the Info Dictionary of the output PDF file. This information is displayed to the user if the “general document info” function is selected in Adobe Acrobat.

- ▶ **Title**

This parameter sets the text for the Title entry in the Info Dictionary of the output PDF file. This information is displayed to the user if the “general document info” function is selected in Adobe Acrobat.

Security

Protecting the content of the PDF document is accomplished with encryption. This PDF security feature is supported by the AFP2PDF transform and follows the password features supported within the Adobe Acrobat product. A PDF document can have two kinds of passwords, a document open password and a Permissions password.

When a *document open password* (also known as a *user password*) is used, any user who attempts to open the PDF document is required to supply the correct password.

The encrypted PDF in the transform is also tied to disabling certain operations when displayed in Adobe Acrobat. Using letter codes, any combination of the following operations listed can be disabled:

- c** Modifying the document's contents
- s** Copying text and graphics from the document
- a** Adding or modifying text annotations and interactive form fields
- p** Printing the document

If any of these operations is disabled, a *permissions password* (also known as an *owner* or *master password*) must also be specified. Any user who needs to override a restricted operation must supply the correct permissions password.

Both types of passwords can be set for a document. If the PDF document has both types of passwords, a user can open it with either password.

The Adobe products enforce the restrictions set by the permissions, owner, or master password.

Note: Not all products that process PDF files from companies other than Adobe fully support and respect these settings. Users of these programs might be able to bypass some of the restrictions set.

The associated parameters in the Option file are:

► **Default_Encryption_Permissions**

This parameter sets the default encryption permissions to be used when encrypting the PDF output. A list of the permission flags include:

- p** Do not print the document from Acrobat.
- c** Changing the document is denied in Acrobat.
- s** Selection and copying of text and graphics are denied.
- a** Adding or changing annotations or form fields is denied.

The following flags are defined for 128-bit encryption (PDF 1.4, Acrobat 5.0):

- i** Disable editing of form fields.
- e** Disable extraction of text and graphics.
- d** Disable document assembly.
- q** Disable high quality printing.

A flag of 5 can be used in combination with one of the “old” flags to force 128-bit encryption without setting any of the i, e, d, or q flags. Using any of these Acrobat 5 related flags produces a file that cannot be opened with older versions of Acrobat.

► **Default_Owner_Password**

This parameter sets a default owner encryption password.

► **Default_User_Password**

This parameter sets a default user encryption password.

AFP2PDF command

As seen before, the conversion function is called automatically from OnDemand. For different purposes, such as creating the image map file or testing the conversion result, you might use the **afp2pdf** command.

See the AFP2PDF transform documentation for details about the **afp2pdf** command parameters.

9.2.3 AFP2XML: converting AFP to XML

AFP2XML is an IBM Services Offering that generates XML documents from AFP files. It contains two modules:

- ▶ **buildICT**

This is a Windows-based GUI used to define XML attribute and value data. Within the interface, the user opens a representative AFP document, selects items on the display, and defines controls on those items.

This module is available on Windows.

- ▶ **afp2xml**

This is a Java executable that generates the XML file from the input AFP file and the control file defined using the buildICT. There is also a user exit available that allows more sophisticated AFP processing.

This module is available on Windows, AIX, Sun Solaris, and z/OS. The Linux and HP-UX versions will be available before the end of 2006. (Contact your IBM representative for more up-to-date information).

The AFP2XML GUI displays the document similarly to its print output. It allows the user to select Triggers and Attributes and define parsing rules without requiring any specific AFP knowledge.

You can extract the significant data from your AFP file and place it into an XML format, which is a more interchangeable format.

The AFP document can be retrieved from OnDemand using one of the ODWEK APIs and then converted to XML using the Java or C API. The XML file can be manipulated for any use of the available information in this interchangeable format.

Consider an example where a company that archives their invoices in OnDemand wants to develop an electronic and bill presentation application, and that their customers can check their bills online and have the opportunity to pay online. The company wants to extract some information from invoices, such as the Amount Due, the Account Number and the Statement Date, to include it in the new online electronic and bill presentation application (see Figure 9-2).

The information is extracted from the archived AFP documents and placed in an XML file so that it can be integrated into the electronic and bill presentation application.

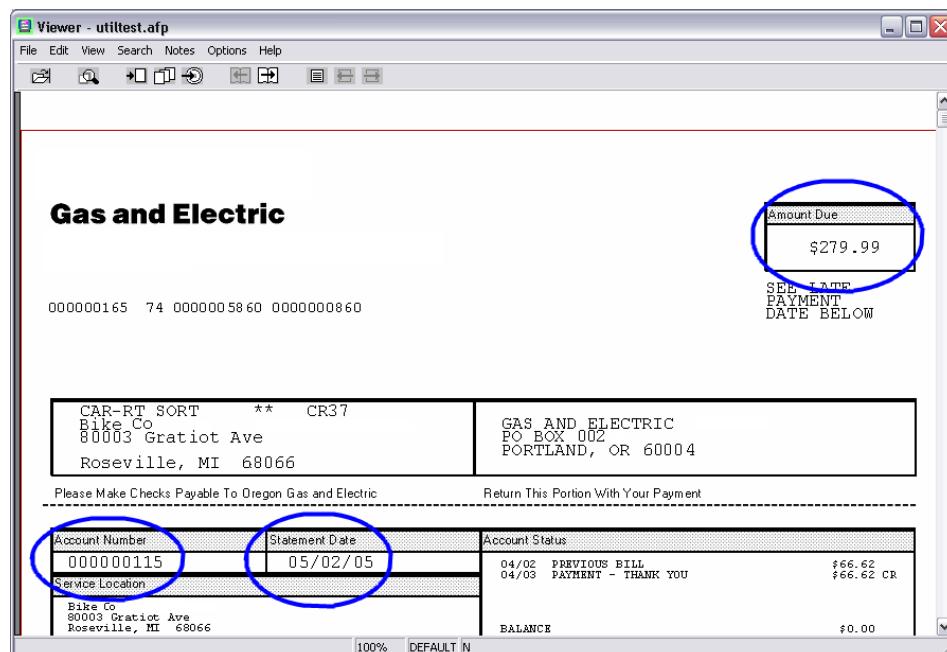


Figure 9-2 AFP invoice

An ICT file is defined using the buildICT client. This control file is used with the afp2xml module to automatically extract information from the invoices and place them into an XML file. Figure 9-3 shows how to define an XML ICT file for AFP invoices.

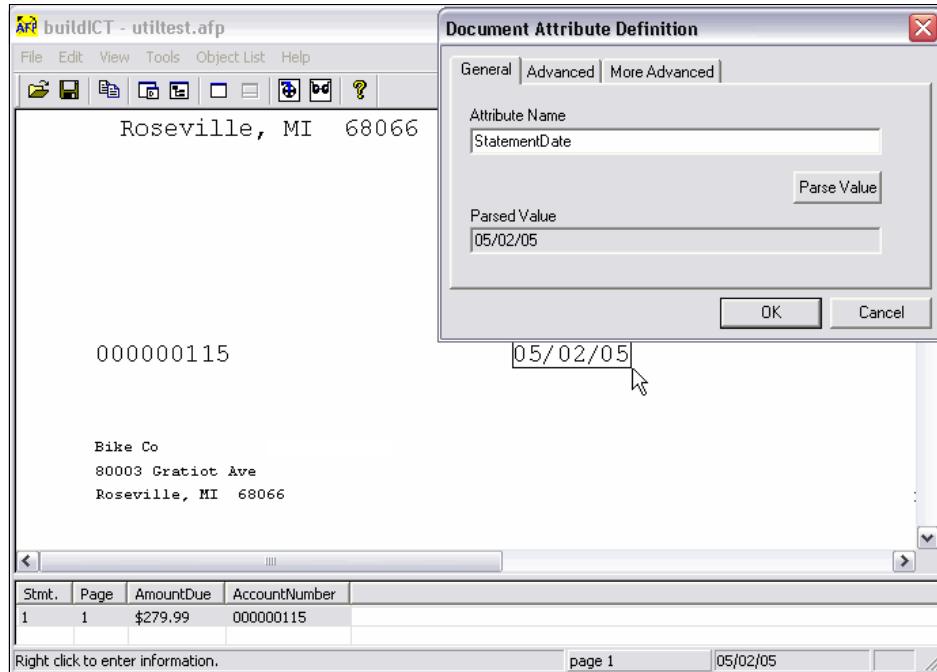


Figure 9-3 Defining an XML ICT file for AFP invoices

The structure of the XML file can be displayed from the buildICT client for control. See Figure 9-4.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Root>
- <Statement>
  <AmountDue>$279.99</AmountDue>
  <AccountNumber>000000115</AccountNumber>
  <StatementDate>05/02/05</StatementDate>
</Statement>
</Root>
```

Figure 9-4 Invoice XML file

9.2.4 AFPIindexer: indexing composed AFP files

It might happen that composed AFP files are generated without the tags that allow the indexation by OnDemand. It is then necessary to process these file in order to insert the requisite tags according to the indexing needs.

AFPIindexer is an IBM Services Offering that generates AFP documents with Page Level Tag Logical Elements (TLE) and Group Level Tag Logical Elements from AFP files and control files.

AFPIindexer contains these modules:

- ▶ buildICT

This Windows-based GUI is used to define Page and Group Level indexing controls. Within the interface, the user opens a representative AFP document, selects items on the display, and defines controls on those items.

This module is available on Windows.

- ▶ indexAfp module

This executable generates the indexed AFP file from the input AFP file and the control file. It generates Page and Group level TLEs in the output file and creates an AFP Document Index file and an AFP Resource Group file.

This module is available on Windows, AIX, Sun Solaris, and z/OS. Linux and HP-UX versions will be available before the end of 2006. (Contact your IBM representative for up-to-date information.)

AFPIindexer generates IBM OnDemand compatible output, an indexed AFP file, AFP Document Index files, and AFP Resource. OnDemand waits for the *.out, *.ind, and *.res files to use them as input for archiving.

The AFPIindexer GUI displays the document in a similar manner to how it displays its print output. It allows the user to select Triggers and Indexes and define parsing rules without requiring any specific AFP knowledge.

Consider an example where a company generates composed AFP invoice files and wants to archive them. The indexes, in this example, should be the Amount Due, the Account Number, and the Statement Date. See Figure 9-5.

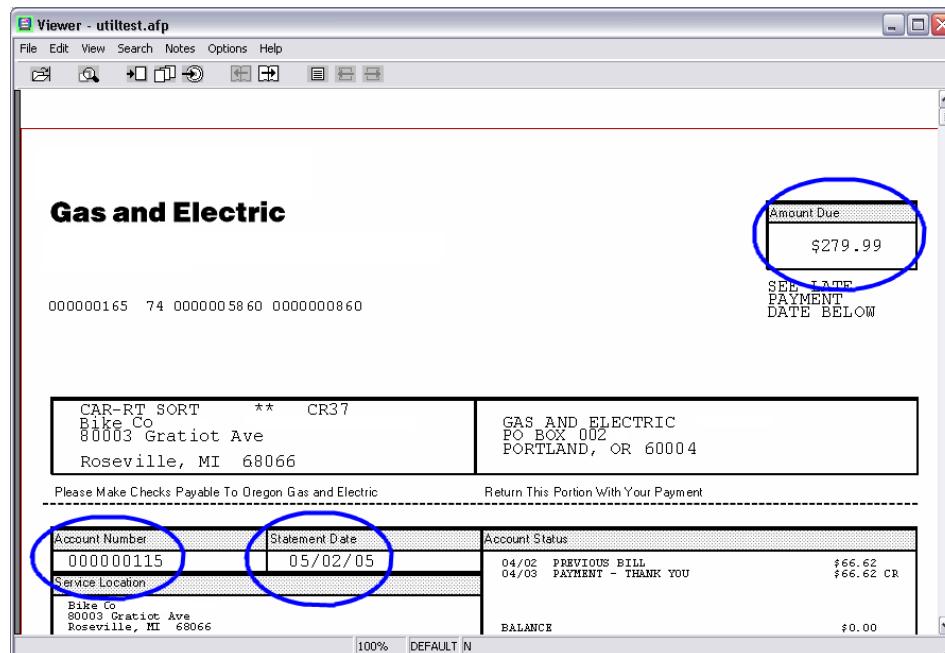


Figure 9-5 AFP invoice

An ICT file is defined using the buildICT client. This control file is used with the indexAfp module to insert the required tag and generate the *.out, *.ind and .res files for archiving by OnDemand. Figure 9-6 shows how to define the XML ICT file for an AFP invoice.

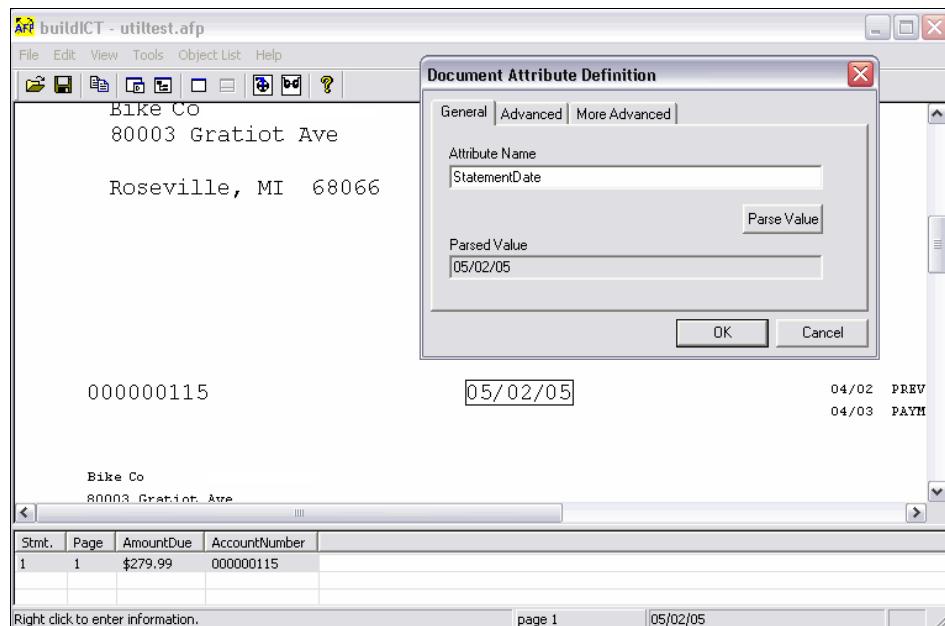


Figure 9-6 Defining an XML ICT file for AFP invoices

The tagged AFP file can be controlled from the buildICT client for validation of the ICT file. See Figure 9-7.

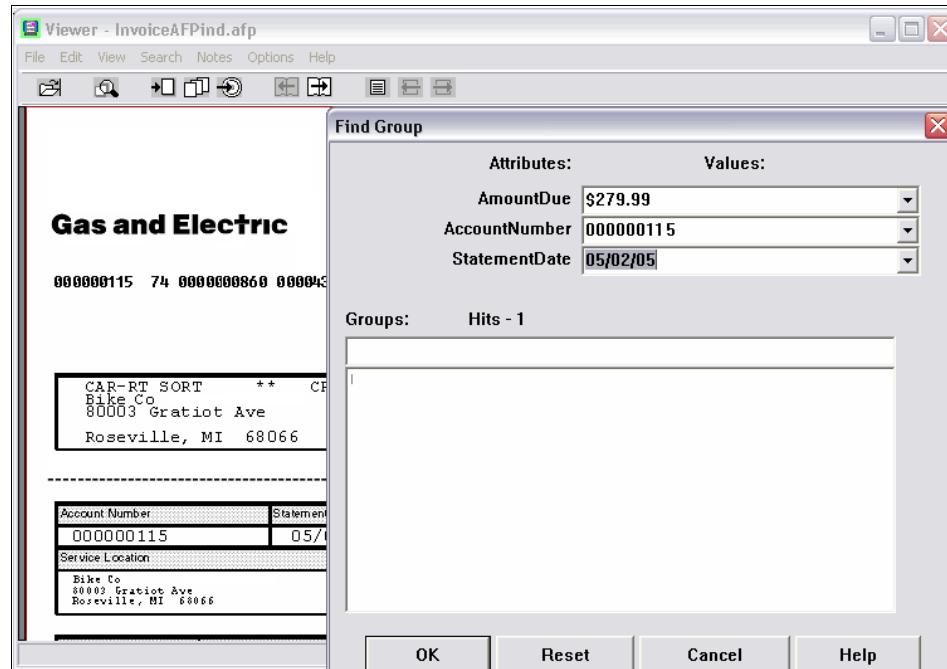


Figure 9-7 AFP document tagged with TLEs for OnDemand indexation

9.3 Xenos and OnDemand

The Xenos d2e platform is a unique component-based solution that transforms data and documents into e-content industry standard formats. The Xenos d2e platform is officially supported with OnDemand Version 7.1 and allows OnDemand to handle data types that were previously unsupported. Xenos d2e allows for the indexing and loading of Xerox metacode (DJDE, LCDS and mixed mode), HP PCL and non-indexable IBM AFP. Additionally, Xenos allows for viewing, through ODWEK, directly through the Web browser. It does this by dynamically converting OnDemand data to e-content formats such as XML, HTML, and PDF.

The Xenos transforms are batch application programs that let you process these various input data types by converting the data, indexing on predefined parameters and collecting resources to provide the proper viewing. The Xenos load transform produces the index file, the output file, and the resource file, which are then used by the **arsload** program to load the data into OnDemand. These documents are then available to users to search, retrieve, and view.

The Xenos transforms can be run either when loading the input files into the system, or alternatively, dynamically when the documents are retrieved via the OnDemand Web Enablement Kit.

If transforming the data at load time, the Xenos transforms listed in Table 9-1 are available.

Table 9-1 Available Xenos transforms: transforming data at load time

From	To
AFP	PDF
Metacode	AFP
Metacode	PDF
Metacode	Metacode
PCL	PDF

If transforming the data dynamically when it is retrieved via ODWEK, the Xenos transforms listed in Table 9-2 are available.

Table 9-2 Available Xenos transforms: transforming data dynamically through ODWEK

From	To
AFP	PDF
AFP	HTML
AFP	XML
Metacode	AFP
Metacode	HTML
Metacode	PDF
Metacode	XML

Figure 9-8 shows, at a high level, how the Xenos transforms fit into the OnDemand environment. It shows the resources and the AFP, metacode, or PCL printstream being fed into the ARSLOAD program. When ARSLOAD runs, it checks to see what indexer to use. If the application specifies Xenos, then the Xenos transforms are called and run with a predefined parameter and script files.

The output files produced by Xenos are handed back to ARSLOAD and the indexes and documents are loaded into the database and storage accordingly. Alternatively, if ODWEK is configured to convert documents at retrieval, then the Xenos transforms are called from ODWEK before presenting the document to the user.

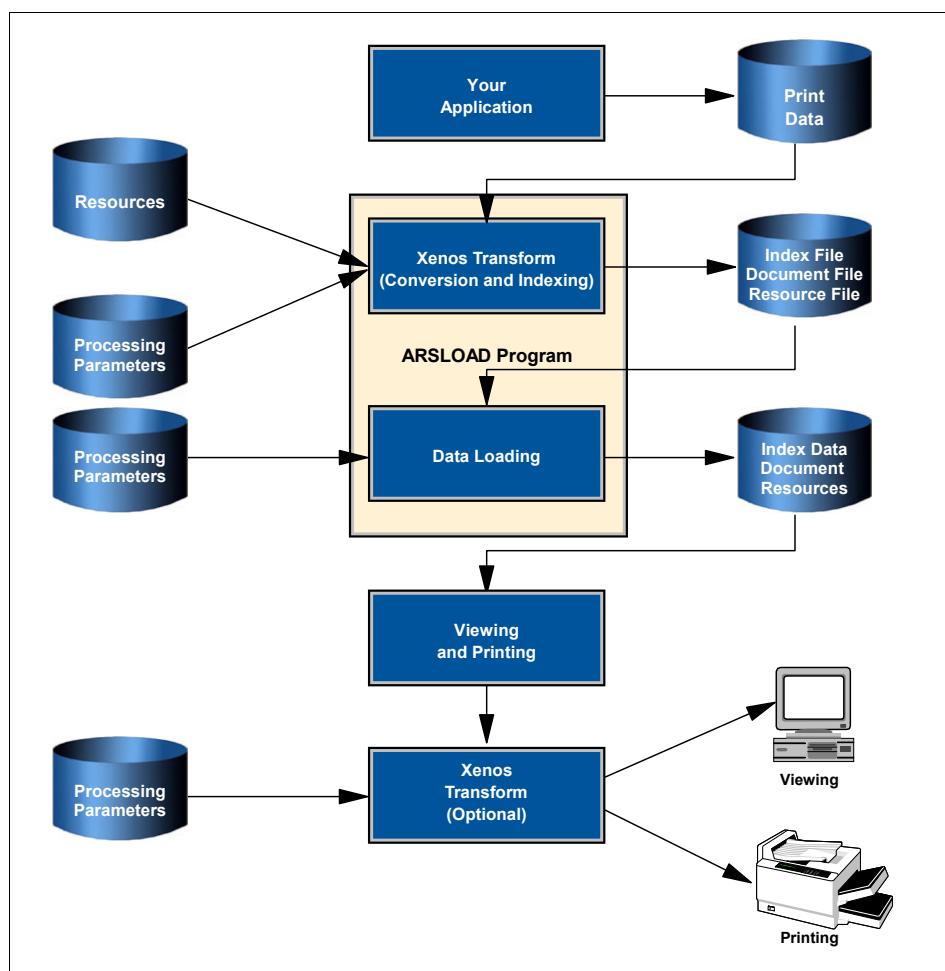


Figure 9-8 How the Xenos transforms fit into the OnDemand environment

9.3.1 Converting AFP to XML through ODWEK

The AFP to XML transform is used when retrieving AFP documents from an OnDemand server through the use of ODWEK. The AFP to XML transform allows data from the input document to be inserted in a predefined template file using the Xenos Template Merger facility. This template file can be a standard XML tagged file format and can be used to display information in different layout than the printed page. The XML file can then be used for online applications such as an online payment system. Multiple templates can be used in a single application to suit the variety of information that is found from page to page in the input document.

In this section, we discuss an example of converting an AFP document to an XML document via the Xenos AFP2XML transform and ODWEK. We walk through the steps that make this transform work.

AFP2XML example

For our example, we use an AFP customer billing statement that is stored in OnDemand. We want our customers to be able to retrieve this document in a Web-ready format without having to rely on the AFP Web Viewer. Our Web developers have stated that if they can extract the pertinent pieces of information in a standard XML format, they can use XSL or CSS to format the document. See Figure 9-9 on page 282 to view the AFP statement as retrieved from the OnDemand PC client.

The fields that we want to extract are highlighted with a box. We want to extract the account ID, the amount due, the start and the end dates, and the usage. We also want to extract the current and previous bill sections and all the charges this are included in these sections.

Because Xenos is doing the transform at the document retrieval time, no additional processing is required on the load side. The document has been defined into an OnDemand application group called *xenos-xml*, with a data type of AFP. The document has been loaded with a pagedef, formdef, overlay, and several custom fonts. The data has only two indexes associated with it, AccountID and EndDate. These indexes do not have to coincide in any way with the Xenos fields that we are extracting. However, keep in mind, the fields that are presented to the user in the ODWEK hit list are the fields that are defined in OnDemand, not Xenos. To ensure that the document was loaded correctly, we viewed it both through a PC client and through ODWEK before we added the Xenos processing.

COMPANY A
123 MAIN ST OFC
ROCKPORT TX 75080 08 A 999 1010 1010B GSW 01-11 01-17 02-12-02

READ 1179 1131 48 .08265

4.0

CURRENT BILLING INFORMATION

	<u>AMOUNT DUE</u>
CHARGE FOR GAS USED JAN 11 TO JAN 17	\$23.07
1% CITY FRANCHISE FEE	\$.23
5% SALES TAX	\$1.16
CURRENT GAS BILLING	\$24.46

PREVIOUS BILLING AND PAYMENT INFORMATION

LAST MONTH'S TOTAL BALANCE	\$131.34
PREVIOUS BALANCE	\$131.34

INTEREST PAID IN 2001 \$.16

\$155.80

CLOSING BILL - EXCLUDES ANY APPLIANCES

NOT APPLICABLE THANK YOU FOR THE OPPORTUNITY TO SERVE YOU.
IF PAYMENT ARRANGEMENTS ARE NEEDED, PLEASE
CONTACT OUR OFFICE IMMEDIATELY. AFTER 30 DAYS
ALL UNPAID ACCOUNTS ARE REFERRED TO AN OUTSIDE
COLLECTION COMPANY.

*00700946491B 0 0 0
SERV ADDR: 08 999 1010 1010B 02-12-02 \$155.80 0
123 MAIN ST #OFC 1
ROCKPORT TX 75080
831010825255 ** B006
COMPANY A
PO BOX 825
ROCKPORT TX 75080

Figure 9-9 AFP billing statement stored in OnDemand

As previously mentioned, we want to extract several fields from the AFP document. After we extract these fields from the AFP document, we want to format them into an XML tagged document with the following format, where *xxx* is the value from the document (see Example 9-16).

Example 9-16 Coding example for extracting and reformatting fields

```
<XML>
  <BILLFILE>
    <BILL>
      <BILLSUMMARY>
        <ACCTID>xxx</ACCTID>
        <AMTDUE>XXX</AMTDUE>
        <STARTDATE>XXX</STARTDATE>
        <ENDDATE>XXX</ENDDATE>
        <USAGE>XXX</USAGE>
      </BILLSUMMARY>
      <CURRENT>
        <ITEM>XXX</ITEM>
        <AMOUNT>XXX</AMOUNT>
        <ITEM>XXX</ITEM>
        <AMOUNT>XXX</AMOUNT>
        ...
      </CURRENT>
      <PREVIOUS>
        <ITEM>XXX</ITEM>
        <AMOUNT>XXX</AMOUNT>
        <ITEM>XXX</ITEM>
        <AMOUNT>XXX</AMOUNT>
        ...
      </PREVIOUS>
    </BILL>
  </BILLFILE>
</XML>
```

In the following sections, we explain how to implement this example.

Xenos job parameter file

To configure ODWEK to run the Xenos transforms, two input and configuration files are necessary, the job parameter file (.par) and the document manipulation script (.dms).

The Xenos parameter file is a text file that contains parser and generator required and optional parameters. Examples of these parameters are the names of input and output files and the location of all document resources, such as fonts, pagedefs and formdefs. Also included in this parameter file is a list of the fields to be pulled from the document and where these fields are located.

Five types of job-related parameters can be defined in the parameter file. They are organized by type and begin with a section heading. The five sections are Job Script, Parser, Generator, Display List, and Distributor. Each of these sections contains many required and optional fields depending on the data type that is being parsed and generated. Refer to *Xenos d2e Platform User Guide*, which comes with the Xenos offering by Xenos Group Incorporated, for a full description of this file.

Table 9-3 provides a parameter file summary, with a description for each parameter section that is applicable to our AFP2XML conversion.

Table 9-3 Parameter file summary

Parameter section	Description
Job Script (JS:)	This section indicates the names and locations of the Xenos d2e Script Library. The Dmsl.lib library is required, and the conversion fails if this library is not defined. This section also defines the variables to be used in the d2e script.
Parser (AFPPARSER:)	This section gives information about the incoming document and how to process it. This section defines where the AFP resources (fonts, pagedefs, formdefs, overlays, and page segments) are located. It also defines font correlation table.
Generator (TMerge:)	This section controls how the new document is generated. The XML generator uses the Template Merge Facility, which scans the template for variable names and then replaces them with variable values from the document. In our parameter file, the PREFIX and SUFFIX parameter tell the template merger the characters that define the beginning and ending of the variable in the template file.
Display List (DLIST:)	This section controls how special features, such as book marking and URL links, and fields are generated. Our display list parameters tell d2e where to locate each field in the input AFP file.

Example 9-17 shows the full contents of our parameter file.

Example 9-17 AFPS2XML_rb.par file contents

```

JS:
FDDMSLIB = 'D:\Program Files\xenosd2e\d2ebin\dmsl.lib'
SCRIPTVAR = ('project_path','D:\d2eDevStudio\AFP2XML\'')
SCRIPTVAR = ('project_resource_path','D:\d2eDevStudio\AFP2XML\resources\'')

AFPPARSER:
CC = YES
FDAFPFONTS = 'D:\d2eDevStudio\AFP2XML\Resources\%s.fnt'
```

```

FDFORMDEFS = 'D:\d2eDevStudio\AFP2XML\Resources\%s.fde'
FDMFCT = 'D:\d2eDevStudio\AFP2XML\AFP2XML.tab'
FDOVERLAYS = 'D:\d2eDevStudio\AFP2XML\Resources\%s.ovr'
FDPAGEDEFS = 'D:\d2eDevStudio\AFP2XML\Resources\%s.pde'
FDPAGESEGS = 'D:\d2eDevStudio\AFP2XML\Resources\%s.psg'
FORMDEF = 'f1mbibio'
PAGEDEF = 'p1mbibio'
POSITION = WORD

TMERGE:
PREFIX = '&&'
SUFFIX = '.'

DLIST:
PARMDPI = 100
PAGEFILTER = ALL
FIELDNAME = 'PAGE'
FIELDWORD = %30
FIELDPHRASE = %400
FIELDLOCATE = ('CurrentBill','BILLING INFORMATION')
FIELDLOCATE = ('EndCurrent','CURRENT GAS BILLING')
FIELDLOCATE = ('EndPrevious','PREVIOUS BALANCE')

FIELDNAME = 'ACCTID'
FIELDBOX = (367,78,519,98)
FIELDWORD = %20
FIELDPHRASE = %500

FIELDNAME = 'AMTDUE'
FIELDBOX = (714,530,816,578)
FIELDWORD = %20
FIELDPHRASE = %500

FIELDNAME = 'STARTDATE'
FIELDBOX = (585,81,641,99)
FIELDWORD = %20
FIELDPHRASE = %500

FIELDNAME = 'ENDDATE'
FIELDBOX = (652,81,714,100)
FIELDWORD = %20
FIELDPHRASE = %500

FIELDNAME = 'USAGE'
FIELDBOX = (730,130,769,148)
FIELDWORD = %20
FIELDPHRASE = %500

FIELDNAME = 'CURRBILL'

```

```
FIELDBASE = ('',41,'',220,'',800,'EndCurrent',30)
FIELDWORD = %20
FIELDPHRASE = %5000
FIELDTABS = (0,450)

FIELDNAME = 'PREVIOUSBILL'
FIELDBASE = ('EndCurrent',-40,'EndCurrent',40,'',800,'EndCurrent',200)
FIELDWORD = %20
FIELDPHRASE = %5000
FIELDTABS = (0,450)

FIELDNAME = 'Field'
FIELDBOX = (176,119,263,163)
FIELDWORD = %20
FIELDPHRASE = %500
```

The job parameter file can either be typed in manually using any ASCII editor or created graphically using the d2e Developer Studio. In most cases, it is a combination of both. Developer Studio has a graphical AFP parser that can render the input file as a bitmap. This allows for the graphical selection of field locations for data extraction. We used the Developer Studio wizard to create the parameter file and locate the fields, and then updated this file manually for the resource locations and the script variables.

Xenos document manipulation script

The document manipulation script, or **dms**, is the second file that ODWEK needs to run the Xenos transforms. The **dms** is a REXX program that is used to customize and enhance the capabilities of the d2e transform program. The script file is used in conjunction with the job parameter file to determine how a particular job is processed. The parameters defined in the script override any defined in the parameter file. Refer to *Xenos d2e Platform Scripting Reference*, which comes with the Xenos offering by Xenos Group Incorporated, for a complete understanding of this script.

In our AFP2XML transform, the **dms** script calls the AFP parser to extract the applicable fields from the document. It then calls the template merge function using a set of predefined template files. The output of this transform is an XML tagged text file with the information converted dynamically from the AFP document that is stored in OnDemand.

See Example 9-18 for our complete **dms** script.

Example 9-18 AFP2XML_rb.dms script file contents

```
/* Document Breaking Script */
/* d2e Developer Studio v5.1.63 */
/* Project : AFP2XML */

/* Initialize IDC engine */
CALL dm_Initialize

/* variables */
TRUE = 1
FALSE = 0
doc_open = FALSE

/* Start parsers and Generators */

AFP_Parser_h = dm_StartParser("AFP")

TMerge1_h = dm_StartGenerator("TMerge")

/* Define input and output to ODWEK */

rc = dm_SetParm(APF_Parser_h, 'fdinput', inputfile);
rc = dm_SetParm(TMerge1_h, 'fdoutput', outputfile);

/* open generator documents */

rc = dm_TMergeOpen(TMERGE1_h, outputfile, project_resource_path || "startfile.tpl")

SAY "Finished starting up process"

/* get page */
dlpage_h = dm_GetDLPage(APF_Parser_h)

SAY "Parsed page " dlpage_h

DO WHILE(dlpage_h <> 'EOF')

/* Get field values */
ACCTID = dm_GetText(dlpage_h,"ACCTID",1)
AMTDUE = dm_GetText(dlpage_h,"AMTDUE",1)
STARTDATE = dm_GetText(dlpage_h,"STARTDATE",1)
ENDDATE = dm_GetText(dlpage_h,"ENDDATE",1)
USAGE = dm_GetText(dlpage_h,"USAGE",1)
```

```

/* test for break and begin new XML Bill*/
IF ACCTID <> previous_ACCTID THEN DO
    IF doc_open = TRUE THEN DO
        rc = dm_TMergeWrite(TMERGE1_h, project_resource_path || "endbill.tpl")
    END
    previous_ACCTID = ACCTID
    doc_open = TRUE
END

/* write out Bill Summary Section */

IF doc_open = TRUE THEN DO

    rc = dm_TMergeWrite(TMERGE1_h, project_resource_path || "bill.tpl")

    /* This section parses through the current charges section detail lines and */
    /* and loads them into the xml templast as the detail ITEM and detail NUMBER */

    rc = dm_GetMultiText(dlpage_h,"currbill",curr_cnt)
    DO i = 1 to curr_cnt.0
        PARSE var curr_cnt.i.dm_textdata item '09'x amount

    /*Merge with Template*/
        rc = dm_TMergeWrite(TMERGE1_h, project_resource_path || "itemizedline.tpl")
    END

    /* write out end of current section and beginning of previous section */

    rc = dm_TMergeWrite(TMERGE1_h, project_resource_path || "endcur.tpl")

    /* This section parses through the previous charges section detail lines and */
    /* and loads them into the xml templast as the detail ITEM and detail NUMBER */

    rc = dm_GetMultiText(dlpage_h,"previousbill",prev_cnt)
    DO i = 1 to prev_cnt.0
        PARSE var prev_cnt.i.dm_textdata item '09'x amount
        rc = dm_TMergeWrite(TMERGE1_h, project_resource_path || "itemizedline.tpl")
    END
END

/* get page */

dlpage_h = dm_GetDLPage(APF_Parser_h)
SAY "Parsed page " dlpage_h

END

```

```

SAY "Finished processing pages, now closing...""

/* End last bill and file template */
rc = dm_TMergeWrite(TMERGE1_h, project_resource_path || "endfile.tpl")

/* Close and Finish */

rc = dm_TMergeClose(TMERGE1_h)

RETURN

```

The **dms** script in Example 9-18 on page 287 refers to several template (.tpl) files. These are the files that are used to create the XML output file. See Example 9-19 for the bill.tpl file that is referenced. Each && begins a variable to be replaced with the actual value from the AFP file before it is inserted into the output XML file.

Example 9-19 Template file bill.tpl

```

<BILLSUMMARY>
    <ACCTID>&&ACCTID.</ACCTID>
    <AMTDUE>&&AMTDUE.</AMTDUE>
    <STARTDATE>&&STARTDATE.</STARTDATE>
    <ENDDATE>&&ENDDATE.</ENDDATE>
    <USAGE>&&USAGE.</USAGE>
</BILLSUMMARY>
<CURRENT>

```

By creating templates with as much variable information as practical, the application runs more efficiently. You can create an XML application where each line in the XML corresponds to a different template. However, the speed of the transform is slowed by many more calls to `dm_TmergeWrite` than are necessary. By making each template write more lines in the output XML file, the amount of time for the transform is reduced. With careful template design, the d2e platform applications might speed up considerably, as much as 40% faster.

Configuring ODWEK

After the transform is set up on the Xenos side, you must configure ODWEK to run the transform. You must make the changes explained in the following sections.

Update the arswww.ini file

The [xenos] section contains two parameters, `InstallDir` and `ConfigFile`. The `InstallDir` parameter specifies where the Xenos d2e code is installed. The `ConfigFile` parameter specifies the location of the configuration file used by the Xenos transforms.

Our [xenos] section is as follows:

```
[xenos]
InstallDir=D:\Program Files\xenosd2e\d2ebin
ConfigDir=C:\IBM HTTP Server\cgi-bin\arsxenos.ini
```

You must also update the *AfpViewing* parameter in the [default browser] section. When ODWEK retrieves an AFP document from the OnDemand server, the value of this parameter determines what action, if any, that ODWEK takes before sending the document to a Web browser. To convert AFP documents to HTML, PDF, or XML output with the Xenos transform, specify *AfpViewing=Xenos* so that ODWEK calls the Xenos transform to convert the AFP document before sending it to a Web browser. The type of output that is generated is determined by the value of the *OUTPUTTYPE* parameter in the ARSXENOS.INI file.

Note: This change affects all AFP data on the system; it is not limited to an application group or folder. If you want to override this, you may specify the *_afp* parameter of the Retrieve Document API.

Our updated *afpviewing* parameter is as follows:

```
[default browser]
AfpViewing=xenos
```

Configuring the ARSXENOS.INI file

The ARSXENOS.INI file provides configuration options for the Xenos transform. You typically configure the ARSXENOS.INI file with options for specific OnDemand applications. However, you can also provide a set of default options. The Xenos transform uses the default options when it converts documents from applications that are not identified elsewhere in the file.

Example 9-20 shows our ARSXENOS.INI file. The first section specifies the application group and the application, separated by a dash. Our application group and application are both named *xenos-xml*.

Example 9-20 The ARSXENOS.INI file

```
[xenos-xml-xenos-xml]
ParmFile=D:\Xenos\AFP2XML_rb.par
ScriptFile=D:\Xenos\AFP2XML_rb.dms
LicenseFile=D:\Program Files\xenosd2e\licenses\dmlic.txt
OutputType=xml
WarningLevel=4
```

```
[default]
ParmFile=D:\Xenos\afp2pdf\sample.par
ScriptFile=D:\Xenos\noindex.dms
```

```
LicenseFile=D:\Program Files\xenosd2e\licenses\dmlic.txt  
OutputType=pdf  
WarningLevel=8  
AllObjects=1
```

The *ParmFile* parameter identifies the full path name of the file that contains the parameters that are used by the Xenos transform to convert the document. This points to the `afp2xml_rb.par` file discussed earlier.

The *ScriptFile* parameter identifies the full path name of the file that contains the script statements that are used by the Xenos transform to create the output file. This points to the `afp2xml_rb.dms` script discussed earlier.

The *LicenseFile* parameter identifies the full path name of a valid license file obtained from Xenos.

The *OutputType* parameter determines the type of conversion that the Xenos transform performs. If the input document is AFP, you can set this parameter to HTML, PDF, or XML. If the input document is metacode, you can set this parameter to AFP, HTML, PDF, or XML. Since we are converting from AFP to XML, our parameter is XML.

The *AllObjects* parameter determines how ODWEK processes documents that are stored as large objects in OnDemand. If you specify a value of 0, then ODWEK retrieves only the first segment of a document. If you specify a value of 1, then ODWEK retrieves all of the segments and converts them before it sends the document to the viewer.

Note: If you enable large object support for very large documents, then your users might experience a significant delay before they can view the document.

The *WarningLevel* parameter determines how ODWEK handles return codes from the Xenos transform. The Xenos transform sets a return code after each document is converted. Use this parameter to specify the maximum return code that ODWEK considers to be good and sends the converted document to the viewer.

Viewing the XML document

After the parameter file, the `dms` script, and all the configuration files are updated, you can select a document and see the Xenos output. When we log on to ODWEK and open the `xenos-xml` folder, we see a hit list of all the AFP documents. When we select a document, ODWEK recognizes the document as AFP and checks the `ARSWWW.INI` file to see how to present the AFP data. The `afpviewing` parameter tells ODWEK to execute a Xenos transform, so ODWEK checks the `[xenos]` section to determine the location of the `ARSXENOS.INI` file. ODWEK then looks at the `ARSXENOS.INI` file to determine the parameter and script files to use.

The Xenos AFP2XML transform is then invoked and the XML document is sent to the browser. You can see the XML output in Figure 9-10.

```
- <XML>
  - <BILLFILE>
    - <BILL>
      - <BILLSUMMARY>
        <ACCTID>999 1010 1010B</ACCTID>
        <AMTDUE>$155.80</AMTDUE>
        <STARTDATE>01-11</STARTDATE>
        <ENDDATE>01-17</ENDDATE>
        <USAGE>4.0</USAGE>
      </BILLSUMMARY>
      - <CURRENT>
        <ITEM>CHARGE FOR GAS USED JAN 11 TO JAN 17</ITEM>
        <AMOUNT>$23.07</AMOUNT>
        <ITEM>1% CITY FRANCHISE FEE</ITEM>
        <AMOUNT>$.23</AMOUNT>
        <ITEM>5% SALES TAX</ITEM>
        <AMOUNT>$1.16</AMOUNT>
        <ITEM>CURRENT GAS BILLING</ITEM>
        <AMOUNT>$24.46</AMOUNT>
      </CURRENT>
      - <PREVIOUS>
        <ITEM>LAST MONTH'S TOTAL BALANCE</ITEM>
        <AMOUNT>$131.34</AMOUNT>
        <ITEM>PREVIOUS BALANCE</ITEM>
        <AMOUNT>$131.34</AMOUNT>
      </PREVIOUS>
    </BILL>
  </BILLFILE>
</XML>
```

Figure 9-10 AFP document presented in the XML format sample

Now, we change the afpviewing parameter in the ARSWWW.INI file back to plug-in. With this change, the AFP document is now presented to the browser in its native format and displayed with the AFP Web Viewer (see Figure 9-11).

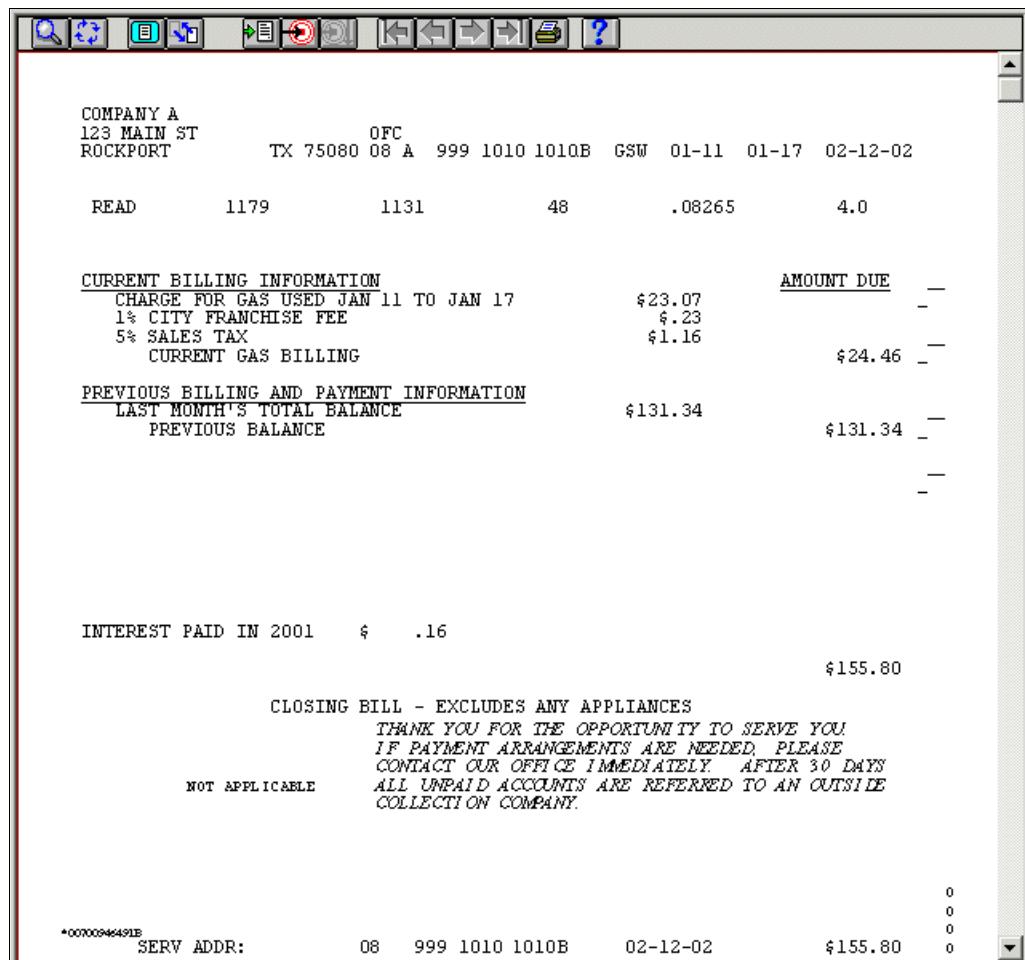


Figure 9-11 AFP document displayed in its native format with AFP Web Viewer

This XML document is presented with standard tags and can be displayed using a variety of XML display methods. We created a simple cascading style sheet (CSS) definition file, and with a few minor changes to the template, this XML file can now be presented to a browser in a Web-ready format.

We created a CSS definition file that takes each parameter from the XML tagged file and presents it to a browser. This file is named *disp_bill.css*. To call this CSS file and tell the browser to associate it with the XML file, we had to make a change to the template file that is called from our **dms** script. To tell the browser to use the *disp_bill.css* file to present the XML file, we added two lines to the *startfile.tpl* template file as follows:

```
<?xml version='1.0'?>
<?xml-stylesheet type="text/css" href="D:\Xenos\disp_bill.css"?>
<BILLFILE>
  <BILL>
```

Now when we click the document in the hit list, the browser sees the first two lines in the XML file that tell the browser to present the document with the style sheet. The document now appears as shown in Figure 9-12.

ACCTID:	999 1010 1010B
AMT DUE:	\$155.80
START DATE:	01-11
END DATE:	01-17
USAGE:	4.0

CHARGE FOR GAS USED JAN 11 TO JAN 17
\$23.07
1% CITY FRANCHISE FEE
\$.23
5% SALES TAX
\$.16
CURRENT GAS BILLING
\$24.46

PAY BILL

LAST MONTH'S TOTAL BALANCE	\$131.34	PREVIOUS BALANCE	\$131.34
----------------------------	----------	------------------	----------

Figure 9-12 Sample document output using XML and CCS

Example 9-21 is the *disp_bill.css* file that we are using to display the document.

Example 9-21 The disp_bill.css CSS file

```
ACCTID
{display: block;
```

```
background-image: url(acctid.gif);
background-repeat: no-repeat;
margin-left: 100px;
margin-right:600px;
background-color: #CCCCCC;
font-size: 16pt;
font-weight: bold;
border: thin ridge;
padding-left: 120px
}

AMTDUE
{display: block;
background-image: url(amtdue.gif);
background-repeat: no-repeat;
margin-left: 100px;
margin-right:600px;
background-color: #CCCCCC;
font-size: 16pt;
font-weight: bold;
border: thin ridge;
padding-left: 200px
}

STARTDATE
{display: block;
background-image: url(strtdate.gif);
background-repeat: no-repeat;
margin-left: 100px;
margin-right:600px;
background-color: #CCCCCC;
font-size: 16pt;
font-weight: bold;
border: thin ridge;
padding-left: 220px
}

ENDDATE
{display: block;
background-image: url(enddate.gif);
background-repeat: no-repeat;
margin-left: 100px;
margin-right:600px;
background-color: #CCCCCC;
font-size: 16pt;
font-weight: bold;
border: thin ridge;
padding-left: 220px
}

USAGE
{display: block;
```

```

background-image: url(usage.gif);
background-repeat: no-repeat;
margin-left: 100px;
margin-right:600px;
background-color: #CCCCCC;
font-size: 16pt;
font-weight: bold;
border: thin ridge;
padding-left: 245px
}
CURRENT { display: block;
margin-top: 80px;
border-top: 2px groove blue;
border-bottom: 2px groove blue;
margin-bottom: 30px;
background-image: url(paybill.gif);
background-repeat: no-repeat;
background-position: 50% 100%}

CURRENT ITEM {display: block;
font-weight: bold}

CURRENT AMOUNT{ display: inline}

PREVIOUS ITEM,PREVIOUS AMOUNT
{ display: inline;
font-weight: bold;
padding-right: 20px;
padding-left: 30px

}

```

Troubleshooting ODWEK and Xenos

By enabling debugging mode in ODWEK, you can view any error or success message that come from ODWEK or Xenos. To turn on logging, make the following changes to the ARSWWW.INI file.

```
[DEBUG]
log=1
logdir=D:\odwek\logging
```

In this example, log=1 turns debugging on (log=0 turns it off) and logdir is the directory where the log file is created. The log file is named *arswww.log*.

9.3.2 Using the AFP2PDF transform with ARSLOAD

In the previous example, we discussed a transform that occurred on the retrieval side. Xenos can also perform the transform during ARSLOAD, before the print file gets loaded to OnDemand. In addition to transforming the data, Xenos can also provide indexing and resource collection capabilities as well as print file segmentation into logical documents or statements.

To enable the Xenos transforms on the load side, you must specify the indexer to be Xenos in the OnDemand application. When ARSLOAD sees an indexer of Xenos, it calls the Xenos transform with the parameter and script files that are specified in the application. Xenos creates three files to be sent back to ARSLOAD, the index file, the output file, and a resource file. ARSLOAD uses these files to update the database and object server.

When using Xenos to parse the input printstream file, it is possible to allow the transform to pull indexes from the file and to split the file into logical documents. When doing this, it is important to define the same database fields in both Xenos and OnDemand. Xenos provides an .ind file that contains each field and the value. If OnDemand receives too many indexes or not enough indexes, or if the indexes are a different data type, the load fails.

This example discusses the conversion of an AFP document to an Adobe PDF format before loading it into OnDemand. This PDF format can be viewed through the OnDemand client and the ODWEK client with the Adobe Acrobat software. We first used Xenos Developer Studio to select field locations from the document and define document splitting rules. We also created a script file (Example 9-22 on page 299) and parameter file (Example 9-23 on page 303). The parameter file defines the parser and generator that we use. It also contains the locations of the AFP resources to be used to convert the file, and the rules for creating the PDF file. Finally, the script defines the data to be used as the index for the document. In this example, we only define one index field, Name.

We created an application and application group called *xenos-pdf*. We are loading AFP data, but because Xenos converts it to PDF before it is loaded, we define the View Information data type to be PDF. In the Indexer Information page, the indexer is defined as Xenos; this directs OnDemand to call the Xenos transforms. When you use Xenos as the indexer, there are only four parameters in the indexing details: Xenos parameter file, Xenos script file, Xenos License file, and warning level.

See our application as set up in Figure 9-13.

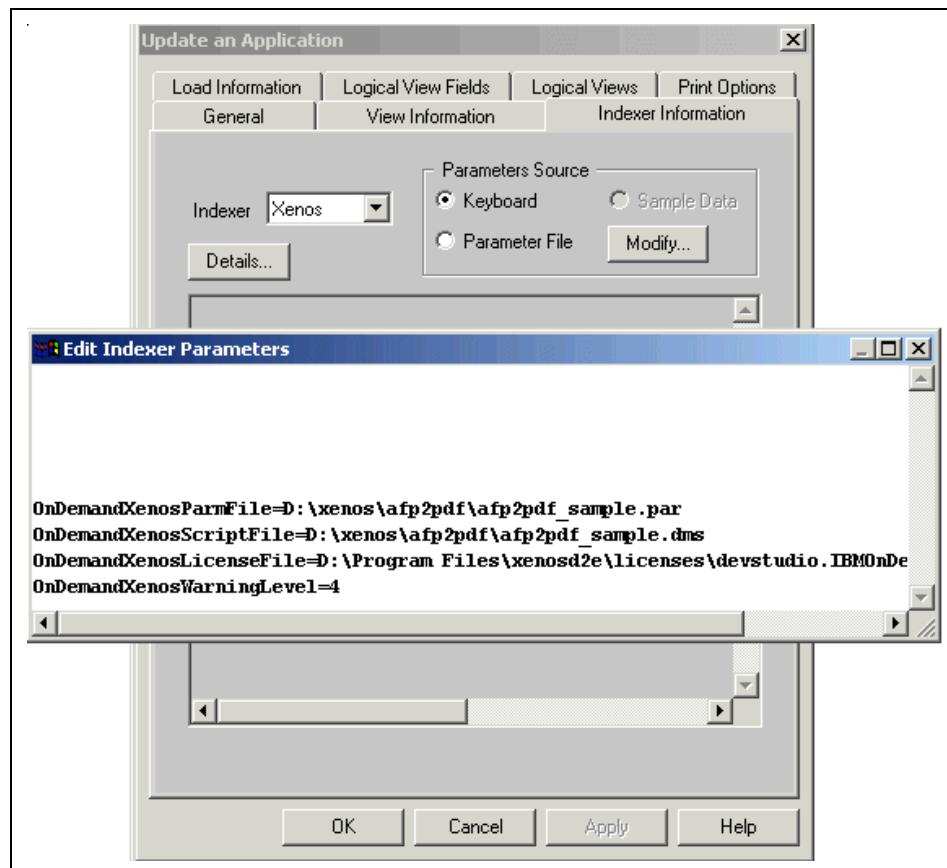


Figure 9-13 Application indexing parameters for Xenos

When defining the Xenos parameter script files in the application, there are two methods: the details method and the filename method. In our example (Figure 9-13), we used the filename method, where we point to the full path of the files. Using the filename method is a way to reuse the same files between multiple applications and is a better method of separating the OnDemand and Xenos administration.

Optionally, you can paste the entire script and parameter file directly into the indexing parameters screen. Using the detail method allows the OnDemand administrator to view the Xenos parameters without the need to access any other system. It is also a way to manage multiple versions of scripts and applications. There is never any question about which application is using which script files. If the printstream data changes, a new application can be created with a new script and parameter file included. When using the detail method, the parameter file details must be included between an *OnDemandXenosParmBegin* tag and an *OnDemandXenosParmEnd* tag. The script details must be included within the *OnDemandXenosScriptBegin* tag and the *OnDemandXenosScriptEnd* tag. Either of these methods works.

When calling the transform from ARSLOAD, be sure not to have any input or output file names hardcoded in the script or parameter file. If you have an input file listed in the fdinput or inputfile parameters, the Xenos transform runs with a return code of 0. It is not processing the data that ARSLOAD is sending. If you have the output files defined in fdoutput, outputfile, indexfile or resourcefile parameters, Xenos also runs fine, but ARSLOAD shows the message, "Output/Indexer file was not created Indexing Failed."

Any error messages that come from the Xenos transforms are populated into the system log and can be viewed in *message number 87*, failed load. All success details that come from the Xenos transform can be viewed in *message number 88*, successful load.

Example 9-22 shows a sample AFP2PDF script file.

Example 9-22 The AFP2PDF_sample.dms script

```
TRUE = 1;
FALSE = 0;

call dm_Initialize

par_h = dm_StartParser(Parser);
gen_h = dm_StartGenerator(Generator);

rc = dm_SetParm(par_h, 'fdinput', inputfile);

/* start the DASD Distributors */
dasd_h = dm_StartDistributor("DASD");
index_h = dm_StartDistributor("DASD");

/* open output and index files */
rc = dm_DASDOpen(dasd_h, '{GROUPFILENAME}'outputfile);
rc = dm_DASDOpen(index_h, indexfile );

/* initialize */
```

```

do i = 1 to NumberOfFields
  fieldvaluesave.i = ""
  if Break.i \= "no" & Break.i \= "NO" then
    do
      Break.i = "yes"
    end
  end
  file_open = FALSE
  save_BytesWritten = 0
  crlf = '0d0a'X

/* write preamble to the index file */
rc = dm_DASDWrite(index_h, "COMMENT: OnDemand Generic Index File Format");
rc = dm_DASDWrite(index_h, "COMMENT: This file has been generated by the xenos process");
dt = DATE('N');
ts = TIME('N');
rc = dm_DASDWrite(index_h, "COMMENT:" dt ts );
rc = dm_DASDWrite(index_h, "COMMENT:" );
rc = dm_DASDWrite(index_h, "CODEPAGE:819"||crlf );

dlpage = dm_GetDLPage(par_h);

do while(dlpage \= 'EOF')
  if file_open = FALSE then do
    select
      when generator = 'PDF' then
        rc = dm_PDFGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
      when generator = 'AFP' then
        rc = dm_AFPGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
      when generator = 'META' then
        rc = dm_METAGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
      otherwise do
        say 'Invalid generator'
        return 12
      end
    end
    if rc = 0 then do
      file_open = TRUE
    end
  end
  end

  do i = 1 to NumberOfFields
    fieldvalue.i = dm_GetText( dlpage, field.i, First )
  end

  docbreak = 0
  do i = 1 to NumberOfFields
    if fieldvalue.i \= "" then do
      /* if there is no previous value, save the current value */

```

```

if fieldvaluesave.i = "" then do
    fieldvaluesave.i = fieldvalue.i
end
else
/* if there is a previous value, see if the new value is different */
if fieldvaluesave.i \= fieldvalue.i then do
    if Break.i = "yes" then
        docbreak = 1
    end
end
end

if docbreak = 1 then do
    select
        when generator = 'PDF' then rc = dm_PDFGenClose( gen_h )
        when generator = 'AFP' then rc = dm_AFPGenClose( gen_h )
        when generator = 'META' then rc = dm_METAGenClose( gen_h )
    end
    file_open = FALSE

/* write out index values to the index file */
do i = 1 to NumberOfFields
    field_name = "GROUP_FIELD_NAME:"||field.i
    rc = dm_DASDWrite( index_h, field_name )
    field_value = "GROUP_FIELD_VALUE:"||fieldvaluesave.i
    rc = dm_DASDWrite( index_h, field_value )
end

/* replace index values with the new values */
do i = 1 to NumberOfFields
    if fieldvalue.i \= "" then do
        fieldvaluesave.i = fieldvalue.i
    end
end

rc = dm_DASDSize(dasd_h)
BytesWritten = dm_size
length = BytesWritten - save_BytesWritten
offset = BytesWritten - length
save_BytesWritten = BytesWritten

group_offset = "GROUP_OFFSET:"||offset
rc = dm_DASDWrite( index_h, group_offset )
group_length = "GROUP_LENGTH:"||length
rc = dm_DASDWrite( index_h, group_length )
group_filename = "GROUP_FILENAME:"||outputfile
rc = dm_DASDWrite( index_h, group_filename||crlf )

select

```

```

when generator = 'PDF' then
    rc = dm_PDFGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
when generator = 'AFP' then
    rc = dm_AFPGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
when generator = 'META' then
    rc = dm_METAGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
end
if rc = 0 then do
    file_open = TRUE
end
end /* end docbreak = 1 */

select
    when generator = 'PDF' then rc = dm_PDFGenWrite(gen_h, dlpage );
    when generator = 'AFP' then rc = dm_AFPGenWrite(gen_h, dlpage );
    when generator = 'META' then rc = dm_METAGenWrite(gen_h, dlpage );
end

dlpage = dm_GetDLPage(par_h);
end

if file_open = TRUE then do
    select
        when generator = 'PDF' then rc = dm_PDFGenClose( gen_h )
        when generator = 'AFP' then rc = dm_AFPGenClose( gen_h )
        when generator = 'META' then rc = dm_METAGenClose( gen_h )
    end
end

/* write out final index values to the index file */
do i = 1 to NumberOfFields
    field_name = "GROUP_FIELD_NAME:"||field.i
    rc = dm_DASDWrite( index_h, field_name )
    field_value = "GROUP_FIELD_VALUE:"||fieldvaluesave.i
    rc = dm_DASDWrite( index_h, field_value )
end

rc = dm_DASDSize(dasd_h)
BytesWritten = dm_size
length = BytesWritten - save_BytesWritten
offset = BytesWritten - length
save_BytesWritten = BytesWritten

group_offset = "GROUP_OFFSET:"||offset
rc = dm_DASDWrite( index_h, group_offset )
group_length = "GROUP_LENGTH:"||length
rc = dm_DASDWrite( index_h, group_length )
group_filename = "GROUP_FILENAME:"||outputfile
rc = dm_DASDWrite( index_h, group_filename )

```

```
rc = dm_DASDClose( dasd_h )
rc = dm_DASDClose( index_h )
return;
```

Example 9-23 shows the corresponding parameter file that is used for Xenos transforms.

Example 9-23 The AFP2PDF_sample.par parameter file

```
/* Xenos Job Parameter file */
```

JS:

```
/* DM Script Library - XG supplied functions */
fddmslib = 'd:\program files\xenosd2e\d2ebin\dmsl.lib'

scriptvar=(‘Parser’, ‘AFP’)
scriptvar=(‘Generator’, ‘PDF’)
scriptvar=(‘NumberOfFields’, 1)
scriptvar=(‘Field.1’, ‘Name’)

AFPDL-AFPP:
/* AFP Parser Options */
formdef = f1a10111
pagedef = p1a06462
CC      = on
trc     = off
startpage = 0
stoppage = 0
native   = no
position  = word

/* File Defs */
FDpagesegs  = ‘D:\xenos\afp2pdf\Resources\%s.psg’
FDafpffonts = ‘D:\xenos\afp2pdf\Resources\%s.fnt’
FDpagedefs  = ‘D:\xenos\afp2pdf\Resources\%s.pde’
FDformdefs  = ‘D:\xenos\afp2pdf\Resources\%s.fde’
FDoverlays  = ‘D:\xenos\afp2pdf\Resources\%s.ovr’

FDfontcor  = ‘D:\xenos\afp2pdf\Resources\master.tab’
FDResGrpOut = ‘D:\xenos\afp2pdf\Output\sample.res’
ResGrpOption = (FormDefs, PageSegs, Overlays)

PDFGEN-PDFOUT:
/* PDF Out Generator Options */
/* output file name being set in the script */
```

```

offset      = (0,0)
scaleby    = 100
border     = NONE
Compress   = (NONE,NONE,NONE)
orient     = AUTO
PDFAuthor  = 'Xenos Group'
PDFOpenAct = '/FitH 800'
BMOrder   = (AsIs,AsIs,AsIs)

AFPDL-DLIST:
parmdpi = 300
pagefilter = all
resfilter = all

FieldName = (PAGE)
FieldWord = (20, and, %20)
FieldPhrase = (%100)
FieldPara = (%500)

/* extract name */
FieldLocate = ('InsName', 'Insured')
FieldName = ('Name')
FieldBase = ('InsName', +275,
             '=', -35,
             '=', +800,
             '=', +30)

FieldWord = (20, and, %20)
FieldPhrase = (%500, OneSpace)
FieldPara = (%500)

```

9.3.3 Job Supervisor program

The Job Supervisor program is the main Xenos transform program. There are three methods for calling this program. The first method allows the ARSLOAD program to call Job Supervisor to transform a complete data file at load time, as discussed in the previous section. This method uses the input file from the **arsload** command and sends the output back to **arsload** when the transform has completed, at which time, OnDemand loads the data.

The second method allows the Web Enablement Kit to call the Job Supervisor program when a document is requested from ODWEK. This calls Job Supervisor with one specific document, allows the document to be transformed, and then sends the transformed data back to the browser. This method is discussed in 9.3.1, “Converting AFP to XML through ODWEK” on page 281.

The third method of calling the Job Supervisor program is from a command line. This might be a useful troubleshooting technique, because it runs Xenos without any connection to OnDemand and allows you to pinpoint any problems. The Job Supervisor program can also be used to print the locations of text strings found on the pages of an input file. These text locations are necessary for defining indexes and locating extraction data. Using Developer Studio is a graphical method of defining data locations and is a much simpler technique.

Developer Studio: Developer Studio allows you to define fields from the document to be used for indexes. Two types of fields are of interest, the *absolute field* and the *relative field*. The absolute field is defined by the x and y coordinates of a box around the text of interest. The relative field is useful for extracting data that has different positions in a page, but can be found in relation to another piece of text on the page. For more information about using Developer Studio, refer to *Xenos d2e Developer Studio User Guide*, which comes with the Xenos offering by Xenos Group Incorporated.

Figure 9-14 shows the syntax when running Job Supervisor from a command line to convert data.

```
►--JS--parms="fileName"--report="fileName"--scriptvar=inputfile="fileName"----->
----->
----->
----->
----->
----->
```

Figure 9-14 Job Supervisor program syntax

Here, -parms is a file that contains the Job Supervisor parameter (.par) file and the Job Supervisor script (.dms) file. All the parameters are required, but you may place the inputfile, outputfile, and resourcefile parameters in the .par or the .dms file instead of in the command line.

We ran the Job Supervisor program from the command line with our parameter file and script file from the previous AFP2PDF example to ensure the index file was being created correctly. We did this before setting up the PDF application group to ensure that Xenos was working correctly before wrapping the ARSLOAD around it. ARSLOAD runs the same command with the same syntax. We ran the command as shown in Example 9-24.

Example 9-24 Running the Job Supervisor program from a command line

```
js -parms="D:\Xenos\afp2pdf\parms_afp2pdf"
    -report="D:\Xenos\afp2pdf\output\sample.rep"
    -scriptvar=inputfile="D:\Xenos\afp2pdf\input\afp2pdf_sample.afp"
    -scriptvar=indexfile="D:\Xenos\afp2pdf\output\afp2pdf_sample.ind"
```

```
-scriptvar=outputfile="D:\Xenos\afp2pdf\output\afp2pdf_sample.out"
-scriptvar=resourcefile="D:\Xenos\afp2pdf\output\afp2pdf_sample.res"
-licensefile=D:\Program Files\xenosd2e\licenses\dmlic.txt
```

Here, our parms_afp2pdf (Example 9-25) contained the .par and .dms file names.

Example 9-25 The parms_afp2pdf file

```
fdjobparm='D:\xenos\afp2pdf\afp2pdf_sample.par'
fddmsscript='D:\xenos\afp2pdf\afp2pdf_sample.dms'
```

This transform creates an .ind file in the format of the Generic Indexer parameter file. This file can then be loaded into OnDemand with the ARSLOAD command.



Report Distribution

In this chapter, we provide information about Report Distribution, an optionally priced feature of OnDemand for Multiplatforms. Report Distribution provides an easy way to automatically group reports and portions of related reports together, organize them, convert the report data into different formats, and send them through e-mail to multiple users or make them available for printing.

In this chapter, we cover the following topics:

- ▶ Introduction to Report Distribution
- ▶ Defining the distribution
- ▶ Hints and tips

10.1 Introduction to Report Distribution

Report Distribution provides many of the same functions as other parts of the OnDemand system such as querying the database for documents, retrieving the documents from various types of storage media, and providing the ability to print them on a server printer. If these functions are already available in OnDemand, why would you want to use Report Distribution? The answer is simple: automation.

Report Distribution automates the process of querying and retrieving documents as well as sending the documents to a printer or to one or more users via e-mail. Not only is the process automated, you can specify when the documents will be delivered. Another benefit to using Report Distribution is that you can select and combine documents from different reports and organize them by defining their order and separating them using banner pages.

Normally when you think of an archival/retrieval system, you think of the large numbers of documents that are stored, but a small number of documents are retrieved. What benefit does automation and scheduling provide?

The biggest benefit is that as reports are loaded into OnDemand on a regular basis, they can be delivered automatically to one or more users soon after 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, a company creates monthly sales reports and archives them in OnDemand. The reports are needed by sales managers to analyze the results and to plan for future sales and inventory. By using Report Distribution, the delivery of the monthly sales report can be automated so that the sales managers receive the report via e-mail once a month as soon as the report is available in OnDemand. Other examples include auditing that is performed on a periodic basis and workflow items such as processing overdue accounts for credit cards, utility bills, or doctors' bills.

The applications for using Report Distribution are endless but the basis for using it is the same; namely documents are loaded on a regular basis and are needed by one or more users as they become available in OnDemand. Let us look at a specific example.

Fictitious Company A (Comp A) is a company that sells artwork and art supplies. Each month a sales report is created for each region and is archived in OnDemand. After the reports are archived, the regional sales managers need a copy of the reports for planning purposes and for restocking inventory.

The Windows client in Figure 10-1 shows a list of monthly sales reports that have been archived in OnDemand. There are two regional sales reports (Midwest and Northwest) for the months of July, August, September, and October.

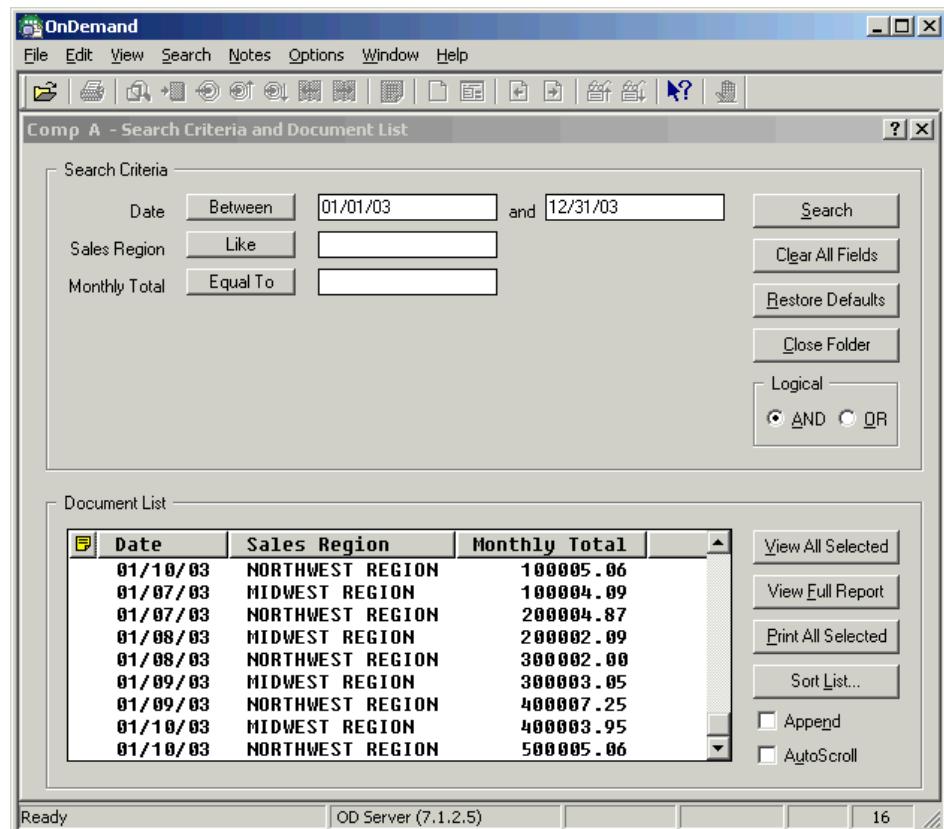


Figure 10-1 List of monthly sales reports for Comp A

In this example, even though there are separate regional sales reports per month, they are loaded at the same time so there is only one load per month. This information is important when you are determining the best way to set up the distribution. Before a distribution is set up, you should ask yourself the following four W questions:

- ▶ What documents are needed?
- ▶ Who will receive the documents?
- ▶ When will the documents be retrieved and delivered?
- ▶ Where will they be delivered?

10.1.1 What documents are needed

For the example, the documents that are needed are the regional sales reports. How do you identify the regional sales reports that you need from the hundreds of thousands of documents stored in OnDemand?

In general, you identify the documents by creating a database query using index fields and values that uniquely identify the documents you want to retrieve. For Report Distribution, another method can be used to identify the documents that you want to retrieve. Instead of querying the database, you can simply retrieve all or some of the documents as they are being loaded.

To illustrate this, let us look at the example again. Once a month, regional sales reports are loaded into OnDemand. Since the load contains all the documents that are needed, we can identify and retrieve all of the documents from the load. Later, when we explain how to set up a distribution using the administrative client, we go into more detail about how to define a set of documents that will be delivered.

10.1.2 Who will receive the documents

The regional sales managers need a copy of the regional sales reports every month. To identify the sales managers to OnDemand, an OnDemand user must be created for each sales manager. Depending on how the documents will be delivered, an e-mail address or a server printer must be specified in the user definition.

10.1.3 When will the documents be retrieved and delivered

Each month, the regional sales managers want to get the regional sales reports as soon as they are available in OnDemand. In Report Distribution, documents can be scheduled for delivery on a daily, weekly, or monthly basis as well as delivering the documents once or delivering the documents a short period of time after they are loaded into OnDemand. For this example, delivery is scheduled based on the availability of the documents after they are loaded.

You might ask why this type of schedule is used rather than a monthly schedule since the reports are loaded on a monthly basis. When a load-based schedule is used, the extraction and delivery of the documents is triggered when the data is loaded. Report Distribution periodically looks to see if data has been loaded. If it has, then the documents are extracted and delivered. When a monthly schedule is used, the extraction and delivery process are performed on a specified day of the month. For some reason, if the data is not loaded by the specified day, the delivery will fail.

The other reason for using a load-based schedule is that the delivery of the documents might be more timely since they are delivered a short time after they are loaded. Depending on when data is loaded and the day of the month that is used for the monthly schedule, there can be several days between the time that the documents are loaded and the time that the documents are delivered rather than a matter of hours or minutes.

10.1.4 Where will they be delivered

The regional sales reports can be delivered to the regional sales managers either via e-mail or to a server printer. In this example, the reports are delivered via e-mail. By using this delivery method, the managers can either view the reports or print them at a later time.

10.2 Defining the distribution

You define report distribution from the Report Distribution component of the administrative client. Figure 10-2 shows an example of an OnDemand server with the Report Distribution component.

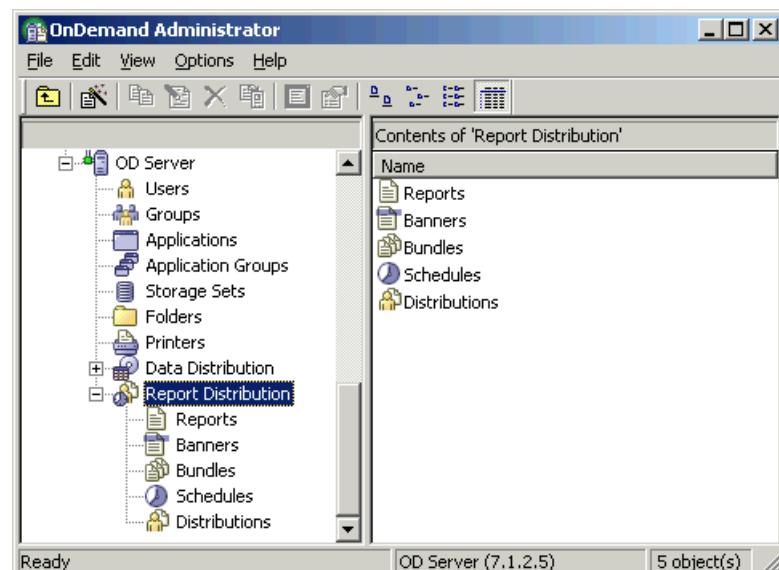


Figure 10-2 OnDemand Administrator Client with Report Distribution

The Report Distribution component contains five subsections:

- ▶ Reports

A *report* provides a way to identify which documents will be extracted from OnDemand. You can identify the documents by creating a database query using index fields and values that uniquely identify the set of documents that you want to retrieve. Alternatively, instead of querying the database, you can retrieve some or all of the documents shortly after they are loaded into OnDemand.

- ▶ Banners

A *banner* is an informational page that is created in the same format as the report data, for example, line, Advanced Function Presentation (AFP), and Portable Document Format (PDF). The content of the banner page is customizable and can include information such as the report name, the report description, and the name of the user that is receiving the distribution.

Three different types of banner pages can be used, a header banner, a separator banner, and a trailer banner. The *header banner* is the first page in the distribution. The *trailer banner* follows the last page of the last report in the distribution. The *separator banner* precedes the first page of every report in the distribution.

► Bundles

A *bundle* is used to specify which reports will be delivered, the order in which the reports will be included in the bundle, the format of the report data, and if banner pages are used, which banner pages will be used. See Figure 10-3 for an example of the contents of a bundle.

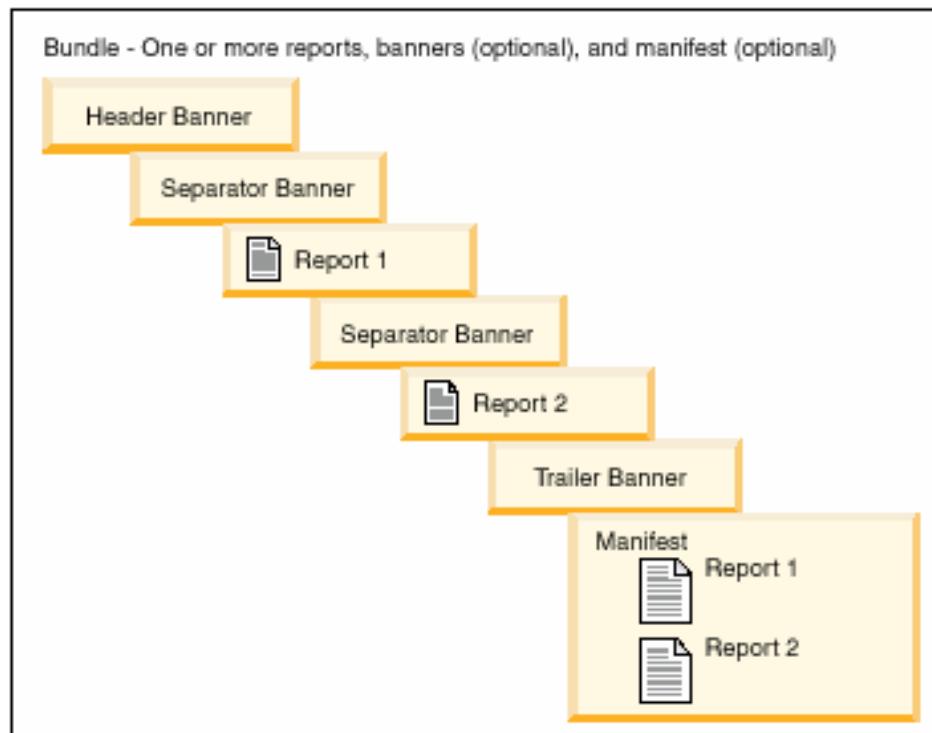


Figure 10-3 Bundle contents

► Schedules

A *schedule* is used to initiate the extraction and delivery of the reports. Reports can be scheduled for delivery on a daily, weekly, or monthly basis as well as one time, or shortly after they are loaded into OnDemand.

► Distributions

A *distribution* is used to identify the bundle of reports that will be delivered, who will receive the reports, when the reports will be delivered, and where they will be delivered.

To set up a distribution, you need the following items, at a minimum:

- ▶ A user
- ▶ A report
- ▶ A bundle
- ▶ A schedule
- ▶ A distribution

This assumes that you have already loaded data into OnDemand so an application, application group, and folder have already been defined. For the example we mentioned earlier, we will include banner pages and multiple reports; the reports will be sent to multiple users.

The logical order to define the distribution objects is to follow the order of the four W questions (who, what, when, and where). Based on this, the order of defining the distribution objects is:

1. Defining users/groups
2. Defining reports
3. Defining banners
4. Defining bundle
5. Defining schedule
6. Defining distribution

Some of these objects might already be defined, so they might also be available for use. For example, users and groups are already used in OnDemand, so you may not have to define these objects. Also, if this is not the first distribution that you are defining, you can use existing banners, bundles, schedules, and reports if appropriate.

For the example, we presume that the users have already been defined and that all sales managers who will receive the monthly regional sales reports already have e-mail addresses specified for their user IDs since they will receive the reports via e-mail. If you do not have to create new user IDs, make sure that the e-mail address or the server printer is specified for each user.

Figure 10-4 shows an example of a user that has an e-mail address and a server printer specified.

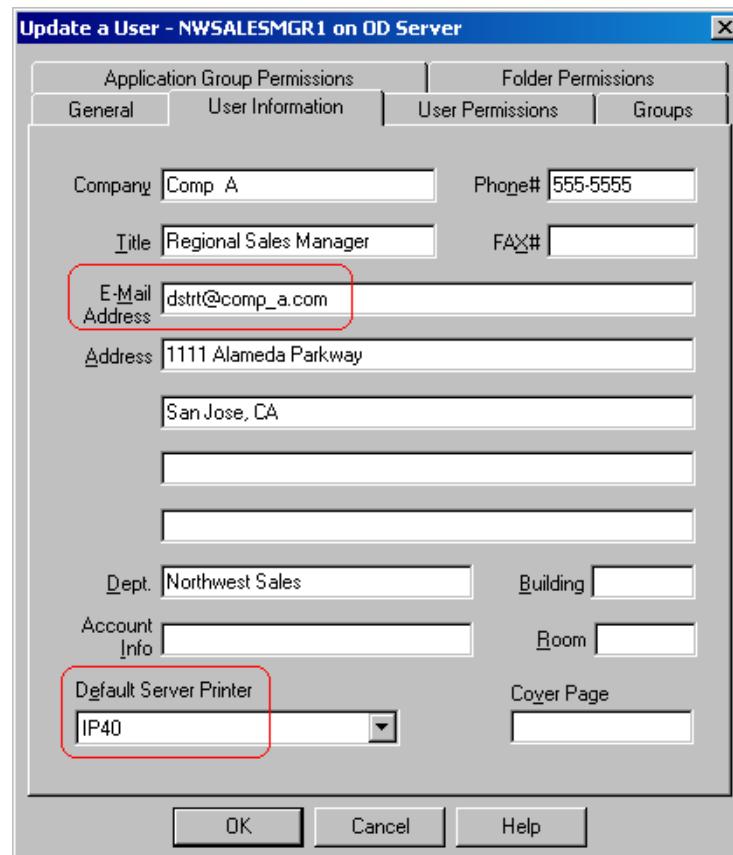


Figure 10-4 User with an e-mail address and server printer specified

10.2.1 Adding a report

The next step is to define the reports to OnDemand. There are three report types: Load, SQL, and Named Query. The *report type* determines the method used to identify which documents will be retrieved:

- ▶ Load

Some or all of the documents are retrieved shortly after they are loaded into OnDemand. When documents are loaded into OnDemand, the load identifier is stored in a database table and a list of documents that have been loaded during the load is created and saved along with the data. The load identifier and the document list are used to retrieve all of the documents for the load. If several loads are processed, there are multiple load identifiers and multiple lists of documents to process for the report.

If you do not want to retrieve all of the documents that were loaded, an SQL query can be used to limit the number of documents to retrieve within the loads that are being processed.

When distributions are set up to be load-driven (for example, reports use the Load report type and the distribution is scheduled using a load-based schedule), Report Distribution periodically checks the load identifier database table to see if there are any load identifiers in the table (meaning data has been loaded). If there is, than Report Distribution begins extracting documents for the report.

- ▶ SQL

An SQL query string is used to query the database. The SQL query string consists of index fields and values that uniquely identify the documents that you want to retrieve.

- ▶ Named Query

A public named query is used to query the database. It contains the database query information that uniquely identifies the documents that you want to retrieve.

To use this method, the named query must have been created prior to defining the report, and it must be a public named query rather than a private named query. A public named query is created from the Windows client.

For the example, two reports are defined, one for the Northwest regional sales report and one for the Midwest regional sales report. Figure 10-5 shows the Northwest regional sales report definition.

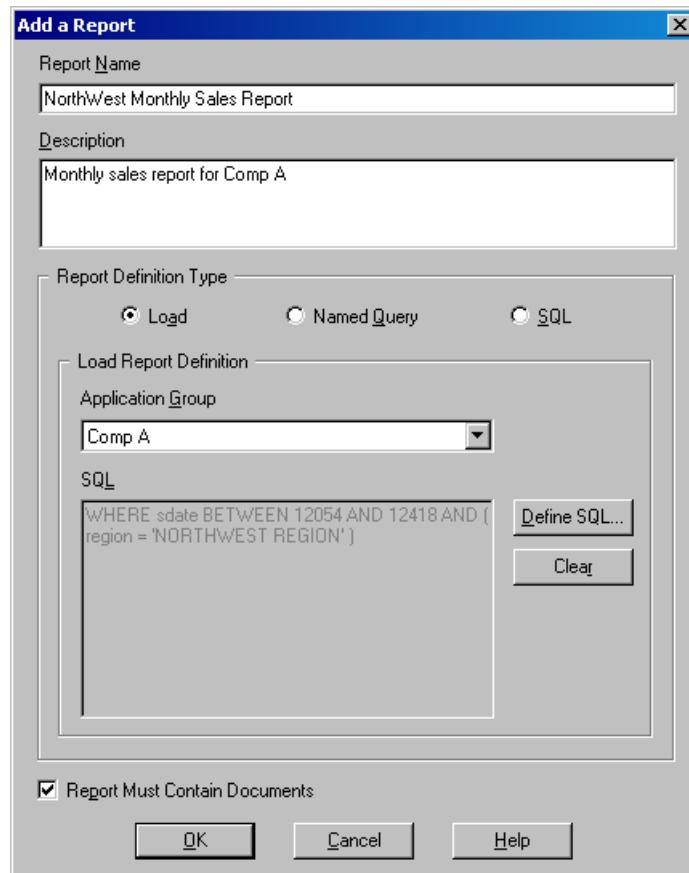


Figure 10-5 Report window for the Northwest regional sales report

For the example, the load report type with an SQL query is used. Since the regional sales reports are in the same load, an SQL query is not necessary. However, by using the SQL query, the reports can be retrieved separately and placed in the bundle in any order. Separator banner pages can also be added to identify each report in the bundle. Another reason for creating separate reports is that it gives you the flexibility to deliver the regional sales report to the appropriate regional sales manager rather than sending both reports. This requires creating a separate bundle and distribution for each report.

Along with selecting the report type, you must select an application group where the data is loaded. The application group must be selected before the SQL query can be defined since the application group database field names are needed to build the SQL query.

In Figure 10-6, an SQL query string is defined for the Northwest region by selecting application group database fields and operators to create the SQL query string. A segment date field can also be specified so that the query is limited to a smaller number of database tables.

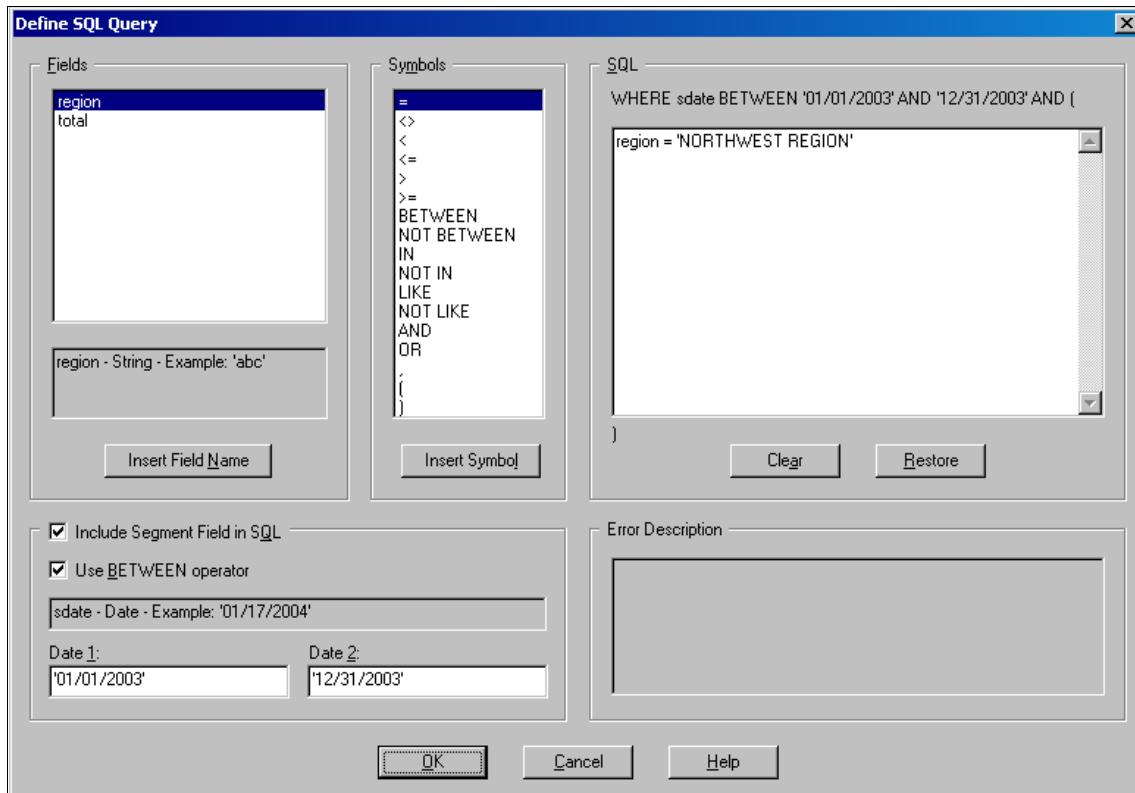


Figure 10-6 SQL Query window for the Northwest regional sales report

After all of the information is entered for the report, click **OK** to save the report. The second report can be created in a similar way. In 10.2.4, “Adding a bundle” on page 321, we explain how to create a bundle.

10.2.2 Adding a banner

You can add header banner, separator banner, and trailer banner to your report. The difference between the various banner pages is their location in a bundle and the information that is contained on the page. Banner pages, regardless of type, are created in the same way.

For the example, a header banner, a separator banner, and a trailer banner are used. Figure 10-7 shows the banner window for a separator banner.

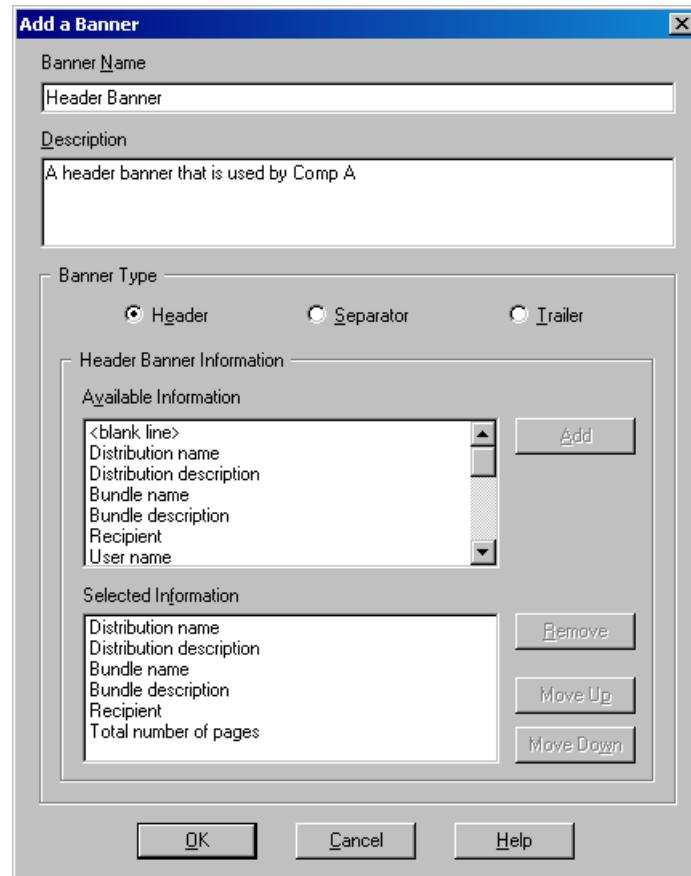


Figure 10-7 Add a Banner window

You must enter the banner name and a description, and then select a banner type. For Banner Type, select the kind of banner that you are creating. After you select the banner type, the list of information that can be included on the banner page is displayed in the Available Information list. You can add one or more lines of information to the banner page. The information can be placed in an order.

The order is defined by the order that you place the information in the Selected Information list. Click **OK** to save the banner. Figure 10-8 shows an example of a separator banner for the Northwest regional sales report.

Report Name: Northwest Monthly Sales Report
Report Description: Monthly sales report for Comp A
Recipient ID: DSTRT
Report Sequence Number: 3
Total pages in the report: 125

Figure 10-8 Example of a separator banner

10.2.3 Adding a schedule

There are five different schedule types: once, daily, weekly, monthly, and load-based. Documents can be scheduled for delivery on a daily, weekly, or monthly basis, or they can be delivered once. Load-based schedules can only be used with reports that are defined using a report type of load.

For the example, a load-based schedule must be used since the reports are defined using a report type of load. Figure 10-9 on page 321 shows the schedule window for a load-based schedule.

As part of the schedule definition, a start date, end date, and delivery time are specified. In the case of a load-based schedule, the Start Date field specifies the first day that you want the Report Distribution to start looking for documents that have been loaded into OnDemand beginning at the time specified in the Delivery Time field. For example, starting on 19 January 2004 at 10:00 AM, Report Distribution periodically queries the load identifier database table to see if any regional sales reports have been loaded.

A Report Distribution parameter determines how frequently it looks for schedules to process. See Figure 10-16 on page 329 for an example of the Report Distribution Parameters window. In the case of load-based schedules, this refers to how often the load identifier database table is queried. After the processing of the load-based schedule begins, it is processed until midnight. At that point, processing of the schedule stops until the next day and starts again at the specified delivery time. If you know what time your data is usually loaded, you can set the delivery time to the same time or a time shortly after that so that Report Distribution does not look for the data before it is loaded. Click **OK** to save the schedule.

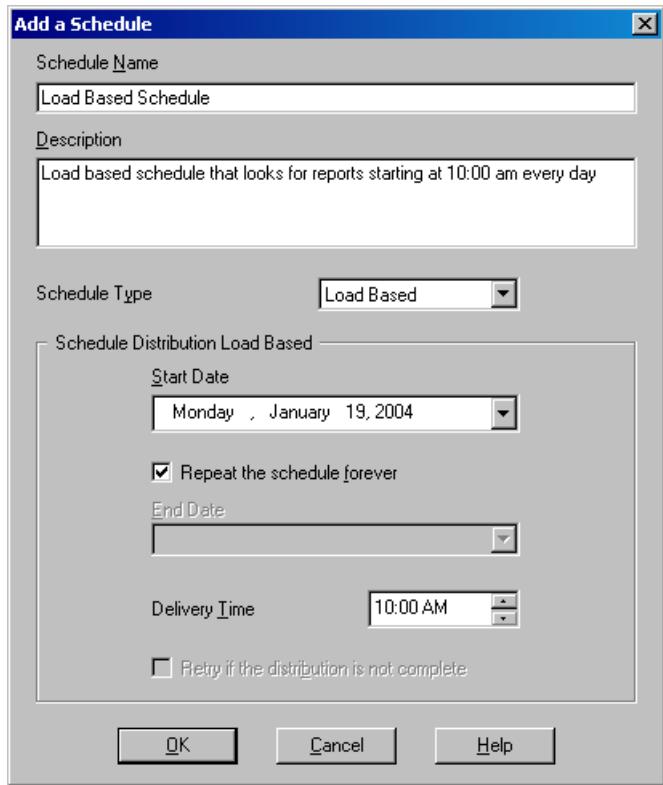


Figure 10-9 Add a Schedule window

10.2.4 Adding a bundle

Figure 10-10 on page 322 shows the General tab for the Add a Bundle window. This is where you specify the output format of the data. For the example, the format of the regional sales reports is ASCII line data so Line Data is selected. If you include more than one report in the bundle, the format of every report must be the same and must match the output format. If the input format is different than the output format, you must use a transform program to convert the report data to the output format. For example, you can use the AFP2PDF transform program to convert AFP reports to PDF. In this case, you specify an output format of PDF.

E-mail notification messages can be sent to one or more users during bundle processing. The message types are error messages, warning messages, progress messages, and completion messages. If you want to notify more than one user, a group can be used. A progress message is sent after the bundle has been processed for each recipient in the distribution. A completion message is sent after the bundle has been processed for all of the recipients.

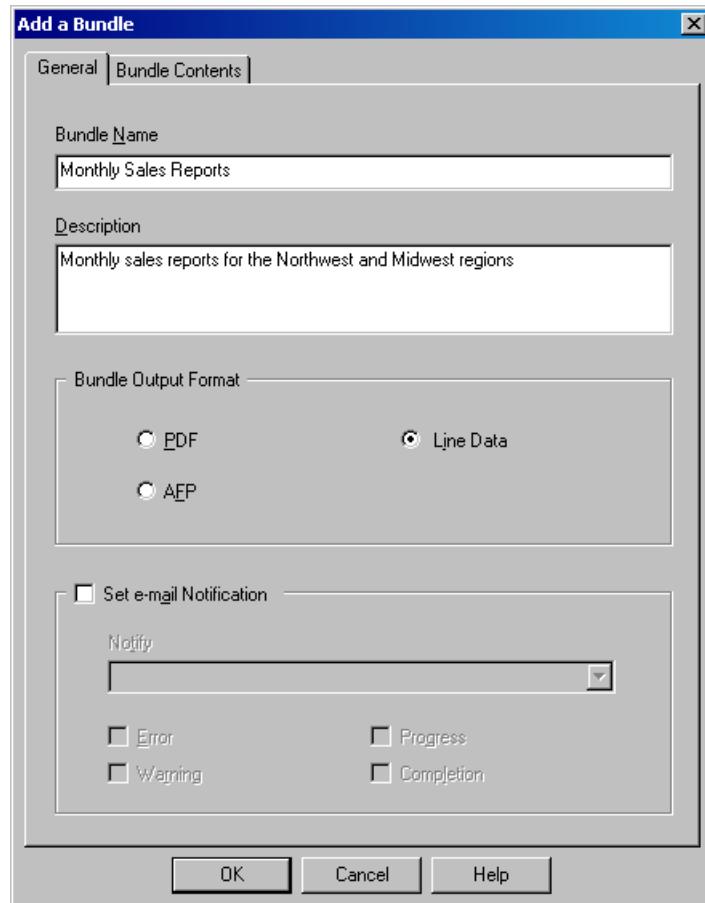


Figure 10-10 General tab of the Add a Bundle window

Figure 10-11 on page 323 shows the Bundle Contents tab of the Add a Bundle window. In this window, you decide which reports to include in the bundle. If banner pages are used, you also specify which banner pages to use.

For the example, the bundle contains a header banner, a separator banner, and a trailer banner as well as the two reports that were created earlier. The first report in the bundle is MidWest Monthly Sales Report and the second report is NorthWest Monthly Sales Report. The reports can be included in the bundle in any order; it is determined by how you add them to the Bundle Contents list. You can change the order of the export by moving them up and down in the list.

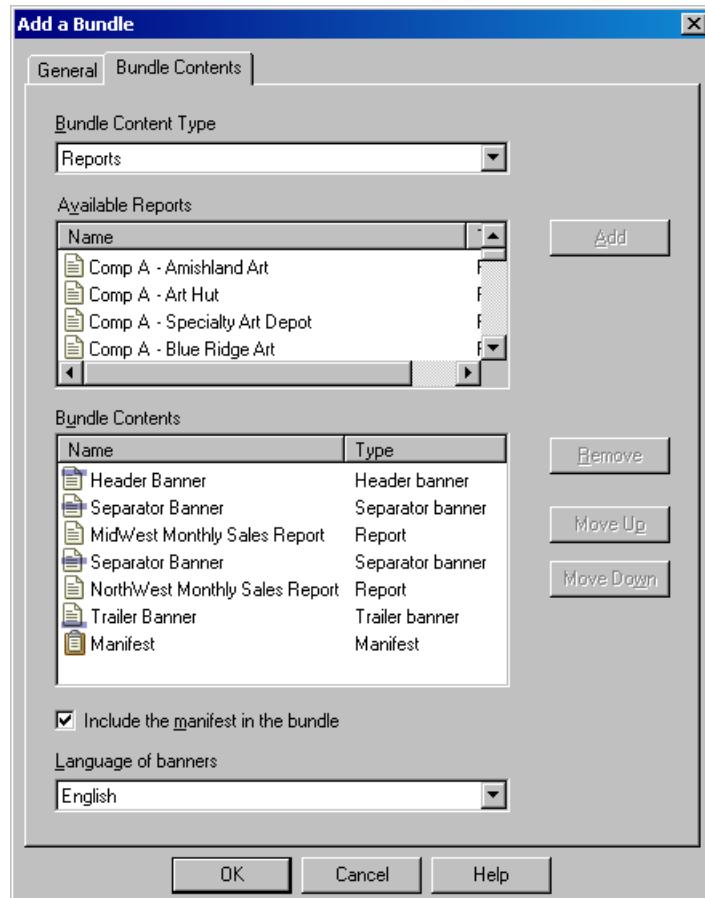


Figure 10-11 Bundle Contents tab of the Add a Bundle window

The inclusion of a manifest in the bundle is optional. The *manifest* is similar to a banner page in that it contains information about the bundle. Specifically, it contains the name of the distribution, the time the distribution is processed, and a list of files containing report data that are included in the bundle. If the manifest is included, it is always added to the end of the bundle.

The field titles on the banner pages and manifest can be created in many different languages. The choices are Arabic, Chinese (Simplified), Chinese (Traditional), Danish, Dutch, English, Finnish, French, French Canadian, German, Italian, Japanese, Korean, Norwegian, Portuguese (Brazil), Spanish, and Swedish. When selecting a banner language, you should consider that the banner pages are converted to the code page of the data. With this in mind, selecting a language, such as Korean, to use with data that is in a single byte code page will not work correctly.

After you make all of the selections for the bundle, click **OK** to save the it.

10.2.5 Adding a distribution

After you create the report, banners, bundle, and schedule, you must put them all together by creating a distribution. Figure 10-12 shows the General tab of the Add a Distribution window. The available Delivery Options are e-mail and server printer. For the example, the delivery method is e-mail.

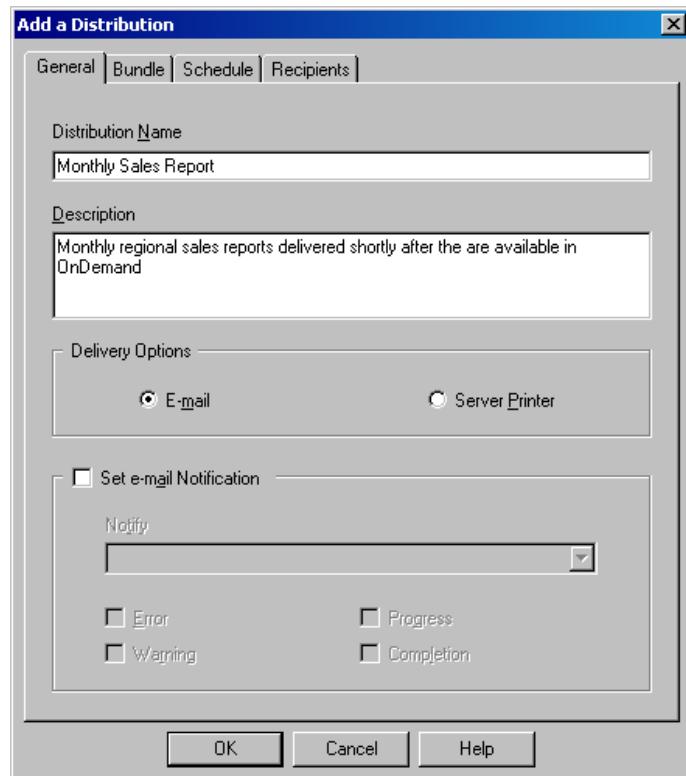


Figure 10-12 General tab of the Add a Distribution window

E-mail notification messages can be sent to one or more users during distribution processing. The message types are error messages, warning messages, progress messages, and completion messages. If you want to notify more than one user, a group can be used. Progress messages are sent after each recipient in the distribution has been processed. A completion message is sent after the distribution has been processed for all of the recipients.

Figure 10-13 shows the Bundle tab of the Add a Distribution window. Only one bundle can be selected for the Distribution Bundle. After the bundle is added, it cannot be changed. For the example, the bundle that was created previously, called Monthly Sales Reports, is selected.

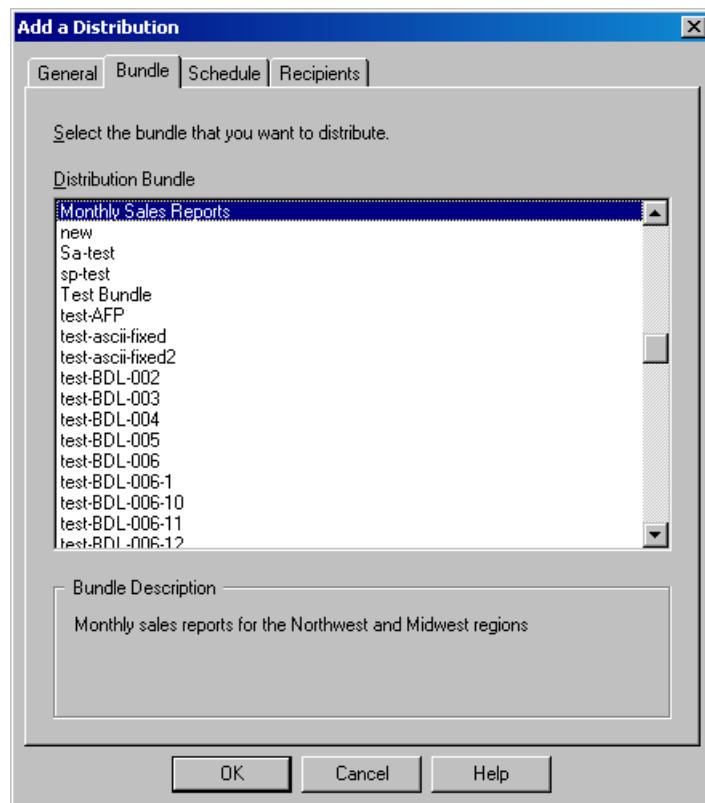


Figure 10-13 Bundle tab of the Add a Distribution window

Figure 10-14 on page 326 shows the Schedule tab of the Add a Distribution window. When you create a distribution, a schedule does not have to be selected or one can be selected but not activated. Of course, if the distribution does not have an activated schedule, the reports in the selected bundle are not delivered.

For the example, we set the Distribution Schedule to *Load Based Schedule* that was created earlier and activate it.

Normally, when a schedule is selected for the distribution, a different schedule cannot be selected. However, if the schedule has expired (for example, today's date is greater than the end date defined in the schedule), Report Distribution removes the schedule from any distributions that use the schedule. If this occurs, a new schedule can be selected for the distribution. The schedule can be deactivated for the distribution at any time.

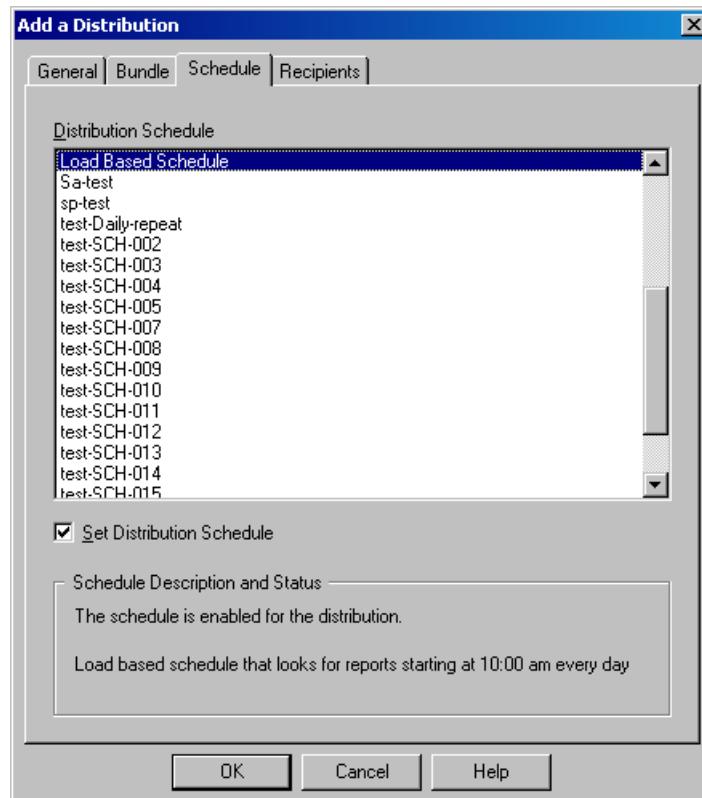


Figure 10-14 Schedule tab of the Add a Distribution window

Figure 10-15 shows the Recipients tab of the Add a Distribution window. In this window, you select the recipients that are going to receive the reports. For the example, the regional sales managers will receive the reports so the user IDs of the managers have been added to the Selected Recipients list. If there are several users that require the same set of reports, you can choose to add the users to a group and add the group to the Selected Recipients list. The two regional sales managers could have been added to a group and then the group would have been used instead of the individual user IDs.

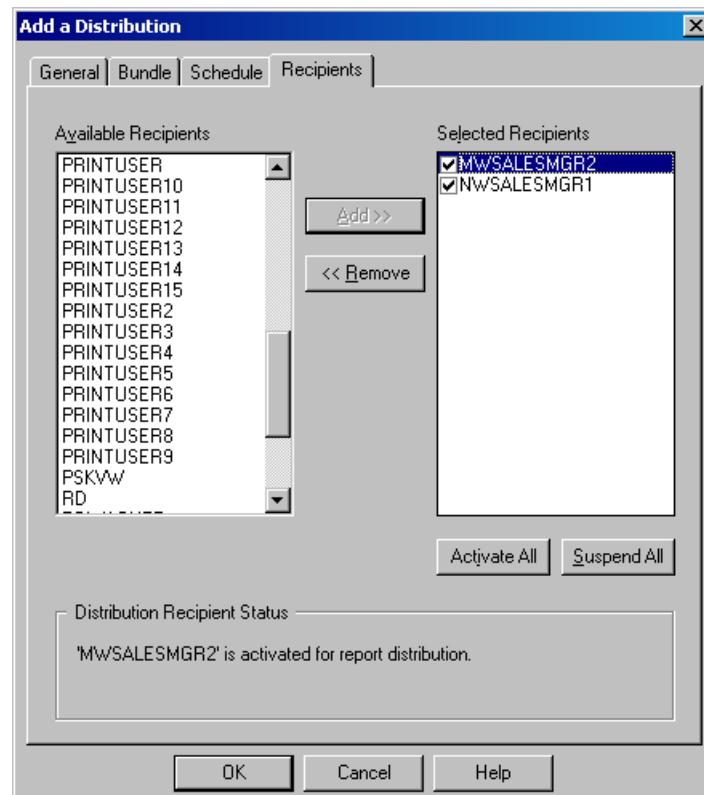


Figure 10-15 Recipients tab of the Add a Distribution window

The check mark next to the user ID of each recipient in the list indicates that the recipient is active for the distribution. If the check mark is removed from the box, the recipient is deactivated and will not receive the reports. You can use this feature to temporarily deactivate the recipient, if for example, the recipient is on vacation and does not need the reports. If there is only one recipient for the distribution, you cannot deactivate this recipient. Also, there must be at least one activated recipient in the distribution and all of the recipients must have e-mail addresses defined if the delivery method is e-mail. If the reports are going to be sent to a server printer, all of the recipients must have a default server printer defined.

After all of the selections are made for the Add a Distribution window, click **OK** to save the distribution. All of the steps required to set up the extraction, bundling, and delivery of the regional sales reports are now completed.

Start the Report Distribution program on the server, and you are ready to begin receiving the regional sales reports after they are loaded into OnDemand.

10.2.6 Report Distribution parameters

There are several parameters that control the Report Distribution process. Figure 10-16 on page 329 shows the Report Distribution Parameters window. To display the window, right-click the **Report Distribution** icon and select **Parameters**.

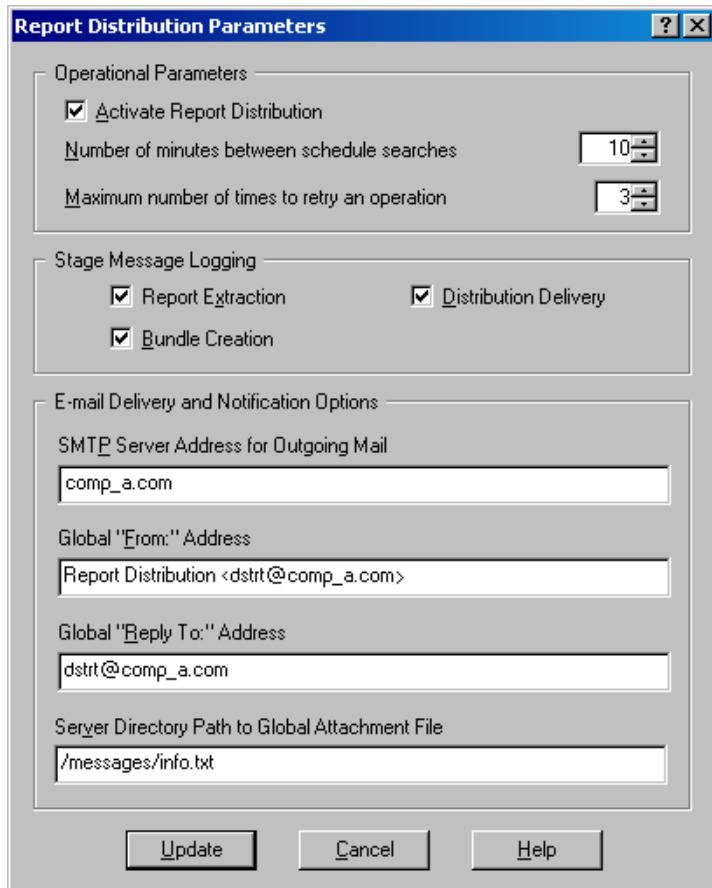


Figure 10-16 Report Distribution Parameters window

One of the Operational Parameters is the Activate Report Distribution. By clearing the check box for this option, Report Distribution can be deactivated so that it temporarily stops processing distributions. Distributions that are in progress will be completed, but no new distributions will be processed until Report Distribution is activated again.

Other options that you can specify include how often Report Distribution looks for distributions that are ready to be processed (Number of minutes between schedule searches) and the number of times an operation should be retried before the operation is marked as failed (Maximum number of times to retry an operation).

Messages that are generated during the extraction, bundling, and delivery stages of Report Distribution can optionally be logged. They are viewable using the Windows client by opening one of the folders that was created during Report Distribution installation.

If you use the e-mail delivery option or need to send e-mail notification messages, you must specify an Simple Mail Transfer Protocol (SMTP) server address that processes the e-mail messages that are generated by Report Distribution. You can optionally specify address information that is specific to your company such as a return e-mail address and an e-mail address to use for correspondence. You can also use a file that contains company-specific information or other types of information that you want to include with the delivery of the documents. If a global attachment file is specified, it is attached as a separate file to the e-mail message that contains the documents that were extracted.

10.3 Hints and tips

To help you successfully work with Report Distribution, we provide the following hints and tips:

- ▶ A *load-based schedule* can only be used with reports that are defined using the *load* report type and vice versa. The load-based method can only be used for data that was loaded using IBM DB2 Content Manager OnDemand for Multiplatforms Version 7.1.1 or later. Data that was loaded with a prior version can be retrieved using the time-based method.
- ▶ After documents are delivered using a load-based schedule, they cannot be delivered again using a load-based schedule. For example, if the January monthly sales report was delivered using a load-based schedule, it cannot be delivered again by a load-based schedule. If for some reason you need to deliver the January sales report again, you can use a schedule with a schedule type of *once*.
- ▶ Consider that you loaded several months of data and now want to start delivering the documents using a load-based schedule starting with the latest month. In this case, you can add a date to the query so that documents from prior months will not be delivered the first time the schedule is processed.

For example, the sales report has been loaded for the last four months and now you want to set up the distribution of the sales report starting with the latest month. You can include information in the SQL query that will start the search for the documents with the latest month. From this point on, reports from new loads will be processed.

- ▶ If a daily, weekly, or monthly schedule is used, use the *Named Query* type reports if new data is loaded on a regular basis and retrieved. A named query lets you set up a date range based on the current date, where an SQL query requires a specific date or date range in the query string. If an SQL query string is used, the dates in the SQL query string must be changed so that only the latest data is retrieved.

For example, if data is loaded on a monthly basis and it is loaded by the third of the month, a monthly schedule can be set up to run on the third of the month and the Named Query specifies a date range of “t - 3d” to “t” for the report date. “t - 3d” means today’s date minus three days. This means that Report Distribution looks for reports that have a report date for the first, second, or third day of the month. For example, to extract the report for the month of January, the query matches documents that have a report date of January 1, January 2, or January 3. Assuming the date in the report data is January 1, the report data for the month of January is extracted and delivered.

- ▶ If a distribution is scheduled for delivery using a *once* schedule and the starting date is in the past, the distribution is delivered immediately (that is, the next time Report Distribution scans for active schedules). For example, if the starting date of the schedule is 22 March 2006 and the current date is 21 April 2006, the distribution is processed immediately.
- ▶ Normally the only way an OnDemand system definition (such as user, group, report, and bundle) is updated is when a user uses one of the administrative programs to update the definition. An exception to this is that, when a schedule expires and it is used in one or more distributions, Report Distribution automatically removes the schedule from all of the distributions that are using the schedule. The schedule is no longer selected in the distribution.

- To determine which distributions are scheduled for delivery, you can use the search option.

You can select the search option by right-clicking the **Distribution** icon in the tree view of the administrative client. When the Search for Distributions window opens, clear the **No Schedule** and **Disabled Schedule** check boxes and click **OK**. Only the distributions that are scheduled for delivery are displayed in the list of distributions. Figure 10-17 shows an example of how to search for scheduled distributions.

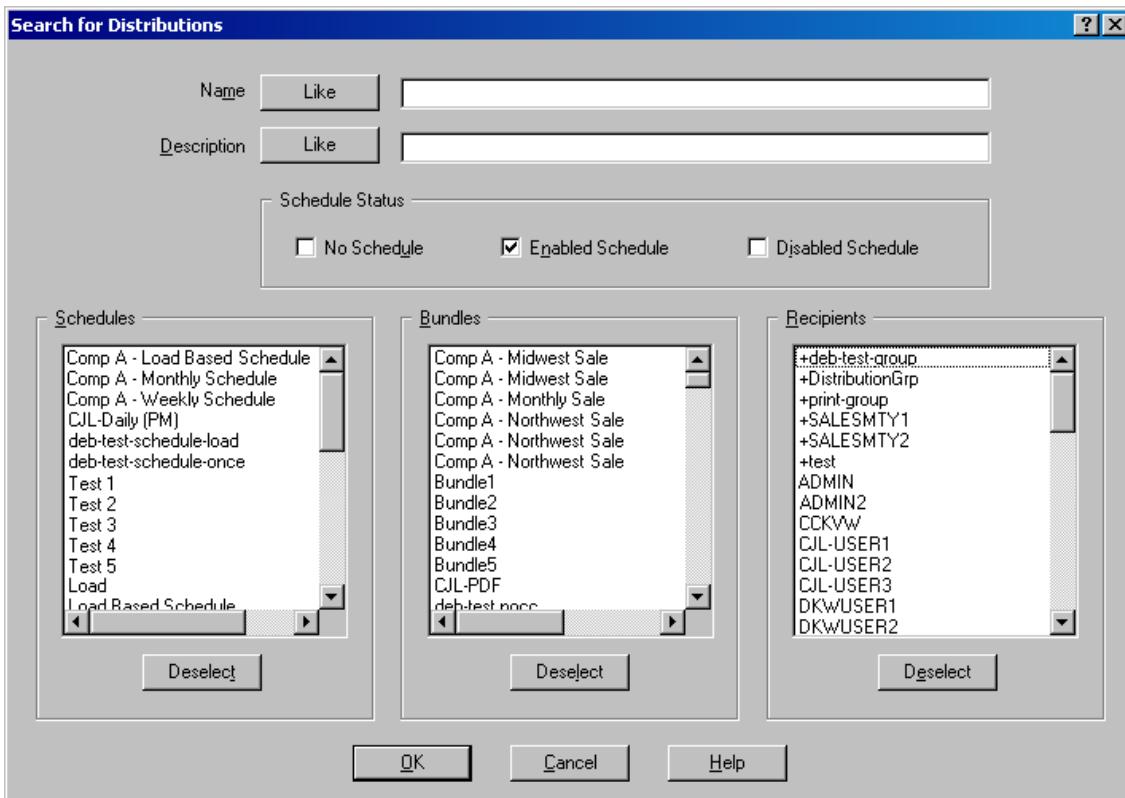


Figure 10-17 Search for Distributions window

- Since the distribution definitions can be updated by the Report Distribution program, you might want to update the distribution information before you search for scheduled distributions.

To refresh the list, select the **Distribution** icon in the tree view of the administrative client and then select **View → Refresh List**. You can also press the F5 key after you select the Distribution icon. The Search for Distributions window also provides a way to identify which distributions use a specific schedule or a specific bundle. You can also identify which distributions will be delivered to a specific recipient.

- ▶ The search option is available for the other Report Distribution areas. For bundles, you can use the search option to determine which bundles contain specific reports or banners. You can use the banner search option to search for banners with a specified banner type, the report search option to search for reports with a specified report type, and the schedule search option to search for schedules with a specified schedule type.



Exits

In OnDemand, it is possible to use exit points to customize and enhance the standard functionality within the product. This chapter introduces a variety of exit points within the OnDemand product. By using actual working sample code, we present some examples of the types of operations and enhanced functions that are possible.

In this chapter, we cover exits in the following areas:

- ▶ Advanced Function Presentation (AFP) Conversion and Indexing Facility (ACIF) exits
- ▶ System administration
- ▶ Customized functions

11.1 Introduction to user exits

A *user exit* is a point during processing that enables you to run a user-written program and return the control of processing after your user-written program ends. There are few different kinds of exits. In this chapter, we discuss the exits based on the following grouping:

- ▶ ACIF Indexing
 - Input record exit
 - Index record exit
 - Output record exit
 - Resource exit
- ▶ System administration
 - System log exit
 - Print exit
- ▶ Customized functions
 - Fax options exit
 - Load exit
 - Permissions exit
 - Preview exit
 - Security exit
 - Storage management external cache exit
 - Tablespace create exit

OnDemand provides data at each exit that can serve as input to the user-written programs. Using these exits, it is possible to perform functions such as e-mailing based on events in the system, updating index values via a print request, cleaning up data as it is loaded into OnDemand, and accessing external security managers. Infinite examples can be provided here for what is possible from the OnDemand exits. We provide some samples here that act as a guide for creating customized user exits programs.

Note: Always make a point to recompile all the customized user exits after upgrading of OnDemand software, because the header files might have changed with different versions.

11.2 ACIF exits

The ACIF user exit is a point during the ACIF processing where control is handed from ACIF to a user-written program. After the user-written program is finished, the control is handed back to ACIF. There are four points during ACIF processing at which user programs can be configured: input, indexing, output, and resource.

Note: ACIF exits are called for each and every input, indexing, output, and resource record. They are not limited to being called only once per file.

In Multiplatforms, ACIF user exits must be written in C. In z/OS, ACIF user exits must be written in COBOL or ASSEMBLER. ACIF exits do not exist in iSeries.

Refer to *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235, and *IBM Content Manager OnDemand for z/OS and OS/390 - Indexing Reference*, SC27-1375, for detailed documentation about each of these exit points.

11.2.1 Input record exit

ACIF provides the input record exit that enables you to add, delete, or modify records in the input file before they are processed by ACIF. The primary purpose of this exit is to be able to modify input records before ACIF sees them.

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 should reflect what the input record looks like *after* the input exit is executed. The only exception is the FILEFORMAT indexer parameter, which should correspond to the input record *before* it is passed to the input exit. For example, if an ASCII stream type file is being loaded, use the FILEFORMAT=STREAM,(NEWLINE=x'0A') parameter, not (NEWLINE=x'25'), an EBCDIC stream delimiter. Otherwise, ACIF does not pass the correct record to the apka2e input exit.

OnDemand provides three input record exits:

- ▶ apka2e
- ▶ asciinp
- ▶ asciinpe

You can either use these 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). There is a much wider selection of EBCDIC coded fonts than there are ASCII, and many customers find it easier to use ones 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 using the **apka2e** exit, you must manually change your indexing parameters:

- ▶ Change CPGID=500.
- ▶ Change the HEX codes for the triggers and fields from ASCII to EBCDIC. If you do not do this, you receive ACIF return code 16, stating that it cannot find trigger1 or any fields.

We used **Hexedit** to determine the new EBCDIC values and typed them by keyboard edit into the parameter file. If you do not have such program, you might find some conversion tables from the Internet.

See 11.2.5, “Debugging user exit programs” on page 342, for further information about how to update indexing parameters.

The **asciinp** exit

The **asciinp** exit program is used when the data does not contain carriage control; rather, it contains “PC style” carriage returns and form feeds X’0D0A’ and X’0DOC’. This IBM provided program transforms the ASCII data stream into a record 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
- ▶ Replaces the ASCII carriage return (X’0D’) with an ASCII new line (X’20’)
- ▶ Replaces the ASCII form feed (X’0C’) with an ASCII new page command (X’31’)
- ▶ Leaves X’0A’ in the file

Note: Because **asciinp** inserts carriage control characters in byte 0 of your document, and leaves X’0A’, it might change 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 performs a combination of the previous two exits. It converts the data from ASCII to EBCDIC and inserts EBCDIC carriage control characters. Refer to the `asciinpe.c` source code for full documentation about this sample program.

11.2.2 Index record exit

The index record exit allows you to modify or ignore the records that ACIF writes in the index object file. The program, specified in the ACIF *idxexit parameter*, receives control just before a record is written to the index object file. The user-written program can tell ACIF to use the record, not to use the record, or to perform some sort of editing on the record before inserting into the index object file.

A good use of this program is for an application that needs to 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`.

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
IDXEXIT( INDEX_EXIT_PARMS *exitstruc )
{
    int i;
```

```

        if ( exitstruc->eof != IDX_EOFLAG )
        {
/*****
/* Look for TLE with attribute name "mmdatyy" */
*****/
        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 30) */
*****/
        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 mmdatyy 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 );
}

```

11.2.3 Output record exit

The output record exit allows you to modify or ignore the records ACIF writes to the output document file. The program is invoked by the ACIF *outexit* parameter, and it gives control to the user program before a record (structured field) 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 tells ACIF not to use this record.

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-APP 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 "acctexits.h"

long
ACCTOUT( 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 );  
}
```

11.2.4 Resource exit

The ACIF resource exit is provided to filter specific resources from the resource file. If you want to exclude a specific type of resource, such as an overlay, you can control this with the ACIF *restype* parameter.

The resource exit is best used to control resources at the file name level. For example, suppose that you intend to send the output of ACIF to PSF and you only want to send those fonts that were not shipped with the PSF product. You can code this exit program to contain a table of all fonts shipped with PSF and filter those from the resource file. The program invoked at this exit is defined in the ACIF *resexit* parameter.

ACIF does not invoke the exit for the following resource types:

- ▶ *Page definitions*: The pagedef is a required resource for processing line-mode application output and 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 it from the *restype* list.
- ▶ *Coded fonts*: If you specify MCF2REF=CF, ACIF includes coded fonts. By default, (MCF2REF=CPCS), ACIF processes coded fonts to determine the names of the code pages and font character sets they reference. This is necessary in creating Map Coded Font-2 (MCF-2) structured fields.

11.2.5 Debugging user exit programs

Sometimes when working with exits, it is necessary to know how the exit has changed your data before you actually load it. A method of doing this 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 defined. Include your input file and the exit routine in the parameter file. Then, run **arsacif** from a command line, pointing to this parameter file. You can also direct the output to a file. Example 11-3 shows our ACIF parameter file, **parmfile**. You use the following command to run stand-alone ACIF:

```
arsacif parmdd=parmfile > 'output filename'
```

This command writes the output of ACIF, including the input exit processing, to the output file where you can inspect it and make sure it 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.

Another method is to run **arsload** with the **-i** option, which runs indexing only. This creates the .ind and .out files for you to view.

Example 11-3 ACIF parameter file

```
CC=NO
CONVERT=NO
CPGID=850
MCF2REF=CPGS
TRC=NO
FILEFORMAT=STREAM, (NEWLINE=X'0A')
DCFPAGENAMES=NO
UNIQUEBNGS=YES
INPUTDD=C:\temp\billing_input.txt
INDEXDD=C:\temp\billing_input.txt.ind
RESOBJDD=C:\temp\billing_input.txt.res
OUTPUTDD=C:\temp\billing_input.txt.out
IMAGEOUT=ASIS
INSERTIMM=NO
RESTYPE=NONE
INPEXIT=C:\Program Files\IBM\OnDemand for WinNT\exits\acif\asciinp.dll
```

To make the debug output more readable, include the INPUTDD, INDEXDD, RESOBJDD, and OUTPUTDD parameters in the parameter file (as shown in the example given).

Important: Specify the *complete path* in the inexit, indexit, resexit, or outexit parameter. There is nothing more frustrating than trying to debug an exit that never gets called because another exit with the same name is being invoked due to the PATH environment variable.

11.3 System administration

In this section, we discuss exits that are used for system administration: system message logging and server printer configuration. These exits are present in the bin directory of the OnDemand installation.

11.3.1 System log exit for Multiplatforms

The OnDemand system log is a tool used by administrators to maintain a log of all of the activity which occurs within an OnDemand server. Each operation performed by a user that involves a connection to the OnDemand server can be logged. The detail that is captured within the system log can be configured so that only certain messages are retained, while others can be discarded.

The system log exit is supplied in the arslog file that resides in the bin directory of the OnDemand install root for each respective platform. If the arslog file is opened in a text editor, notice that it simply contains comments that provide a brief description of the exit and the order of the parameters that OnDemand hands to this exit. By default, the system log exit is not initialized within OnDemand. Therefore, if you edit the arslog file to capture information, the exit is not executed automatically.

To activate the system log exit:

1. Start the administrative 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.

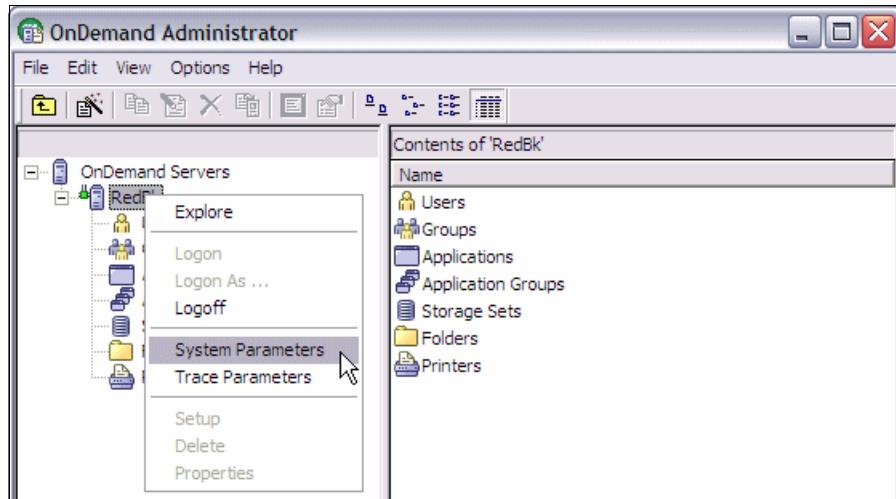


Figure 11-1 Select OnDemand system parameters

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 is calling this exit. A common reason for getting no response from this exit is *access permissions* on either the arslog file itself or files and directories that are being accessed within **arslog**.

OnDemand provides an exit for each of the four system logging event points. These exits allow you to filter the messages and take action 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 some of the most common uses for the system log exit, we provide three typical examples:

- ▶ Capturing failed logon attempts (AIX)
- ▶ Sending an e-mail when a load fails (Windows)
- ▶ Notifying another system when a load has completed (AIX)

For simplicity, we have not demonstrated the system log exits across all supported platforms. We recognize that the scripting languages between platforms do 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 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 some 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 provided in Example 11-4, we have also provided a small sample of what the output of this exit might look like as in Example 11-5. For instance, you can see in the output provided that several unsuccessful attempts have been made from the same machine and different user ID have been used at each attempt. In this example, by adding parameter 2 (\$2) to the output and resorting the file, we can further establish the time of these attempts.

Example 11-5 Sample exit output

```
ADMIN Failed login: GB55102K3.boulder.ibm.com 9.17.46.21
MARTIN Failed login: GB55102K3.boulder.ibm.com 9.17.46.218
FRED Failed login: GB55102K3.boulder.ibm.com 9.17.46.218
USER1 Failed login: GB55102K3.boulder.ibm.com 9.17.46.218
USER2 Failed login: GB55102K3.boulder.ibm.com 9.17.46.218
USER3 Failed login: GB55102K3.boulder.ibm.com 9.17.46.218
```

Sending an e-mail when a load fails (Windows)

On Windows machines, the system log exit file is called *arslog.bat*, rather than *arslog* due to the batch file naming convention used in the Windows environment. Example 11-6 is a system log exit extracted from a Windows machine that collects a variety of information when the exit receives *message number 88* (the message code for a failed load process). In this example, when the necessary information is collected and stored to the *failedload.txt* file, then the file is e-mailed to the OnDemand administrator. The e-mail program used in this case is the same command line e-mailer that is shipped with the E-mail Notification services offering although any e-mail program is sufficient.

Example 11-6 Sending an e-mail when a load fails (Windows)

```
REM
REM %1 - OnDemand Instance Name
REM %2 - Time Stamp
REM %3 - Log Identifier
REM %4 - Userid
REM %5 - Account
REM %6 - Severity
REM %7 - Message Number
REM %8 - Message Text
REM
if (%7)==("88") echo Message - %8 > c:\temp\failedload.txt
REM =====
REM == Message number 88 is a failed load ==
REM =====
echo Time - %2 >> c:\temp\failedload.txt
echo UserID - %4 >> c:\temp\failedload.txt
echo Account - %5 >> c:\temp\failedload.txt
echo Severity - %6 >> c:\temp\failedload.txt
"d:\program files\ibm\ondemand for winnt\email\arssendmail" -b
c:\temp\failedload.txt -f ondemand-server@ibm.com -h d06m1032 -s %8 -t
administrator@ondemand.com
REM =====
REM == arssendmail is a simple command line e-mail tool ==
REM =====
fi
```

Notifying another system when a load has completed (AIX)

This sample was used in a live production environment where the number of load jobs that were sent to OnDemand needed to be controlled so that the next load job was only sent when the previous one completed successfully. In this case, this is because there was a limited amount of disk space in the location on the OnDemand server where the load files were received from the remote machine and that the load files were extremely large.

Example 11-7 shows how the exit collects virtually all of the available information when it received *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 11-7 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
```

Important: For a guide about the codes for each of the message types logged in the system log, refer to Chapter 2, “Common Server Messages”, in *IBM Content Manager OnDemand - Messages and Codes*, SC27-1379. For example, message number 88 is listed as ARS0088I.

11.3.2 System log exit for z/OS

OnDemand can be configured to record information, warning, and error messages. You can set up OnDemand to record these messages using the system log exit named the ARSLOG installation exit. The implementation of the system log exit on z/OS is different from those on Multiplatforms. Like other z/OS exits, it uses the MVS Dynamic Exit Facility.

The configuration of the system log exit is done with the administrator client in the Systems Parameters window (see Figure 11-2).

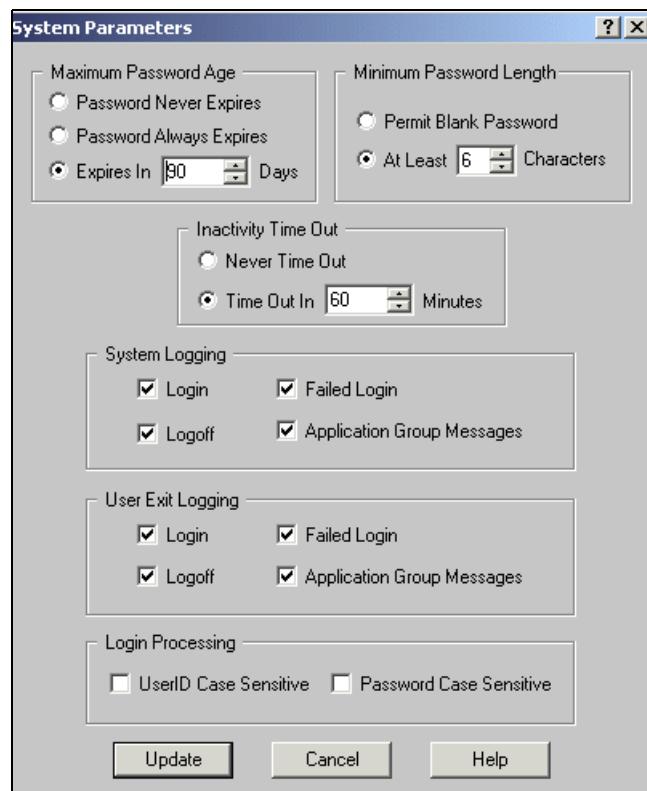


Figure 11-2 System Parameters window for user exit select

Make the selections for the system logging and set up the exit. The sample in Example 11-8 routes the messages to the system log with the WTO Macro.

Example 11-8 System log exit setup sample

```
ARSLOG title 'Issue a message to syslog'          00010000
***** START OF MODULE SPECIFICATIONS *****
*
*
*
* ==> OD/390           - 5655-H39 <==      * 00050007
*
* Module Name:      ARSLOG                      * 00060000
*
* Descriptive Name: Issue a message to syslog   * 00070000
*
* Status:           Version ? Release ?        * 00080000
*
* Function:          This routine issues a message to the SYSLOG * 00130000
*
* Copyright:         5655-H39 (C) Copyright IBM Corp. 2000      * 00140000
*                   Licensed Materials-Property of IBM          * 00150007
*                   See Copyright instructions.                  * 00160000
*
* Notes:             * 00170000
*                   * 00180000
*
* Restrictions:     None                         * 00190000
*
* Register:
*   Convention: R1 points to the Parameter list      * 00210000
*               R12 base register                    * 00220000
*
* Patch Label:      PSPACE                       * 00230000
*
* Input:            Parameter list pointed to by Register 1    * 00240000
*                   Parameter list contains addresses of:       * 00250000
*                   - message length                     * 00260000
*                   - message text                      * 00270000
*                   * 00280000
*
* Output:           None                          * 00290000
*
* Return codes:
*
*   NORMAL:          R15 = return code from WTO        * 00300000
*                   * 00310000
*                   - message length                     * 00320000
*                   - message text                      * 00330000
*
*   Exits:            Return to caller via BR 14        * 00340000
*                   * 00350000
*
* External References:                            * 00360000
*                   * 00370000
*
* Change Activity: See below                      * 00380000
*                   * 00390000
*
*                                         * 00400000
*                                         * 00410000
*
*                                         * 00420000
*                                         * 00430000
*
*                                         * 00440000
```

```

*
*      Ver Rel Mod      Date      Description of Change
*
*      0?  0?  00  04/05/00  Release ??
*
***** END OF MODULE SPECIFICATIONS *****
ARSLOG    csect          * 00450000
ARSLOG    rmode any       * 00460000
ARSLOG    amode 31        * 00470000
               using *,r15   * 00480000
               b    pastcopy    * 00490000
               dc   C'ARSLOG &sysdate' 00500000
               dc   C'5622-662 (C) COPYRIGHT IBM CORP. 2000' 00510000
               dc   C'ALL RIGHTS RESERVED' 00520000
               dc   C'LICENSED MATERIALS-PROPERTY OF IBM' 00530000
               00540000
               00550000
               00560000
               00570000
               00580000
               00590000
pastcopy  ds   0h          00600000
               stm  14,12,12(r13) 00610001
               lr   r12,r15        00620000
               lr   r2,r1          00630000
               using plist,r2      00640000
               drop  r15          00650000
               using ARSLOG,r12    00660000
               storage OBTAIN,length=work1,loc=ANY,cond=YES 00670000
               ltr  r15,r15        00680000
               jnz  bagit          00690000
               st   r13,4(,r1)      00700000
               st   r1,8(,r13)     00710000
               lr   r13,r1          00720000
               using workarea,r13 00730000
               00740000
*
* Determine the message length
*
               slr  r1,r1          Number of bytes 00750005
               l    r15,msgtxta     get starting address 00760000
nulloop   ds   0h          00790006
               cli  0(r15),x'00'    Is it zero? 00800005
               je   nomore         Yes - quit 00810005
               la   r1,1(,r1)       Bump count 00820005
               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	01180005
	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,wto1	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

	lr r1,r13	02270000
	l r2,4(r13)	02280002
	storage RELEASE,length=work1,addr=(r1)	02290003
	lr r13,r2	02300002
	drop r13	02310000
		02320000
bagit	ds 0h	02330000
	lm 14,12,12(r13)	02340001
	br r14	02350000
psize	equ ((*-ARSLOG+99)/100)*5	02360000
	dc C'PATCH AREA - ARSLOG &sysdate'	02370000
pspace	dc 25s(*)	02380000
	org pspace	02390000
	dc ((psize+1)/2)s(*)	02400000
		02410000
		02420000
mvcins	mvc 0(0,r14),0(r15)	02430000
		02450000
wtol	wto text=,	+02460000
	desc=(6),	+02470000
	mcsflag=(BUSYEXIT),	+02480000
	routcde=(11),	+02490000
	mf=L	02500000
wtol1	equ *-wtol	02510000
	ltorg	02520005
		02530005
workarea	dsect	02870000
rsa	ds 18f	02880000
wtoe	ds cl(wtol1)	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 ,	03030007
	iezwp1	03040005
	end ,	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 PROGXX parmlib member. Refer to *z/OS MVS Initialization and Tuning Reference*, SA22-7592, for more information about the PROGXX parmlib member.
- ▶ Use the SETPROG EXIT operator command. Refer to *z/OS MVS System Commands*, SA22-7627, for more information about the SETPORG EXIT command.

You use the following command to activate the exit routine:

```
SETPROG EXIT,ADD,EXITNAME=ARSLOG,MODENAME=ARSLOG, DSN=TEAM5.LOADLIB
```

The exit was link-edited to a normal library that is not AFP authorized.

11.3.3 Print exit for Multiplatforms

There are two ways to print a document stored in OnDemand: *local printing*, via a LAN attached PC printer, or *server printing*, via a printer managed by the print manager installed on the OnDemand server machine. The print exit for Multiplatforms can only be used for documents that are printed via a *server printer*.

The print exit for Multiplatforms is supplied in arsprt file, which resides in the bin directory of the OnDemand install root for each respective platform. If the arsprt file is opened in a text editor, notice that it contains comments that provide a brief description of the exit and the order of the parameters that OnDemand gives to this exit.

Example 11-9 on page 355 shows an arsprt file, which updates application group indexes for a certain document type each time it is sent to a server printer. This is a real example from a customer where the requirement is for OnDemand to keep a record of when a document is reprinted. This file is achieved is 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 that is incremented to log the number of times that the document has been reprinted. Comments are inserted into the sample script in Example 11-9 on page 355 that explain what each part of the script is doing. The customer name and the IP addresses have been either altered or removed for reasons of confidentiality.

Example 11-9 Sample arspt print exit file

```
#!/bin/ksh
#
#      arsprt - OnDemand User Exit Printing Facility
#
# 5622-662 (C) COPYRIGHT IBM CORPORATION 1995
# 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 licensee of the OnDemand product is free to copy, revise,
# modify, and make derivative works of this program sample
# as they 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 that 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 to update   #
# same index several times as only #
# one .cntl file is created even   #
# 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 filname 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=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

```

11.4 Customized functions

The user exits provide customized ways of performing tasks in OnDemand. You can use it to customize logins, define more complex access permissions to documents, retrieve data from external location, or send a notification when a document is loaded. Programming of the user exits is a services offering. You can contact the services team for details. You may also use the sample exit source code to write your own exits. In this section, we discuss each of the sample exits provided in the standard OnDemand installation to give you a better understanding of what they can do.

The sample source code for the OnDemand user exits are provided for all the platforms. They are placed in the exits directory of the OnDemand install root for Multiplatforms. As listed in Table 11-1, these sample user exit modules provide a skeleton for you to program the exits.

The header file provides information about how to turn on the user exits. If it is not specified in the header file, then place the compiled user exit program into the bin/exits directory of the OnDemand installation root. For AIX, the directory is /usr/lpp/ars/bin/exits.

The source code must be compiled before use. For UNIX platforms, you can compile the source code using the sample Makefile that is provided. The Makefile is in the same exits directory as the sample exits source code.

Table 11-1 User exits module

Module	Function	Usage
arsufax	FAXOPTS	Builds fax options based on a particular document and prefilled it when a user faxes

Module	Function	Usage
arsuload	LOADEXIT	To obtain load information for notification
arsuperm	PERMEXIT	Determines folder, application group, and document access and overrides the permission that is defined
arsuprep	PREPEXIT	To preprocess document data prior to document retrieval
arsusec	SECURITY	Validates user IDs and passwords
arsusmxm	SMEEXTCAC	To retrieve document data stored in an external cache
arsutbl	TBLSPCRT	To customize creation of tablespace, tables, and indexes

11.4.1 The user exit header file arscsxit.h

Before you start to write the user exit, it is important to study the header file arscsxit.h. This file is in the same exits directory as the sample user exit source codes. It contains the structure and function declarations for all the OnDemand user exits. There are also instructions on how to activate the user exits after it has been compiled.

The first part of the header file is a declaration of all the structures and variables used. Example 11-10 shows some of the common structures used in the functions declarations.

Example 11-10 Common structure defined in the arscsxit.h header file

```
*****
/* COMMON STRUCTURES
*****
#define ARCCSXIT_MAX_SRVR_MESSAGE_SIZE 1024

typedef struct _ArcCSxitApplGroup
{
    char *name;
    long agid;
    char *agid_name;
} ArcCSxitApplGroup;

typedef unsigned char 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 unsigned char ArcCSxitDocFormat;
#define ARCCSXIT_DOC_FORMAT_FIXED    (ArcCSxitDocFormat) 0x00
#define ARCCSXIT_DOC_FORMAT_VARIABLE (ArcCSxitDocFormat) 0x01
#define ARCCSXIT_DOC_FORMAT_STREAM   (ArcCSxitDocFormat) 0x02

typedef unsigned char ArcCSxitCarCtl;
#define ARCCSXIT_CC_ANSI           (ArcCSxitCarCtl) 'A'
#define ARCCSXIT_CC_MACHINE         (ArcCSxitCarCtl) 'M'
#define ARCCSXIT_CC_NONE            (ArcCSxitCarCtl) 'N'

typedef unsigned char 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;
    long              aid;
    ArcCSxitDocType doc_type;
    ArcCSxitDocFormat doc_fmt;      /* Document Format for Linedata */
    union
    {
        int      fixed;          /* Fixed - Record Length */
        char    stream[17];       /* Stream - Character Delimiters */
    } u;
    unsigned char    trc_present;   /* 0 = no, 1 = yes */
    int              line_count;    /* Lines per page for line data */
    int              code_page;     /* Code Page for line data */
    ArcCSxitCarCtl cc_type;       /* CC type for line data */
    ArcCSxitPrMode prmode;        /* PRMode for line data */
} ArcCSxitAppl;

typedef unsigned char ArcCSxitFieldType;
#define ARCCSXIT_FIELD_TYPE_BIGINT (ArcCSxitFieldType) 0x42

```

```

#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 unsigned char 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
    {
        double      d;
        ArcCSXitBigInt b;
        long        i;
        short       n;
        char        *str;
    } u;
} ArcCSXitField;

typedef struct _ArcCSXitDocFields
{
    int          flds_num;
    ArcCSXitField *flds;
} ArcCSXitDocFields;

#define ARCCSXIT_DOCNAME_SIZE 11

typedef struct _ArcCSXitDocHandle
{
    char          name[ARCCSXIT_DOCNAME_SIZE + 1];
    unsigned long doc_off;
    unsigned long doc_len;
    unsigned long comp_off;
    unsigned long 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. There are structures that are specific to a function itself that are also included in the header file.

In the following sections, we examine each exit and describe its usage and functionality.

11.4.2 Fax options exit

The fax options exit is used by the OnDemand client. When a user chooses to fax a document, the fax options exit can help to prefill the fax options accordingly. Depending on the code, information can be prefilled according to the document being opened.

The fax options exit has a special structure in the header file as shown in Example 11-11, the *ArsCSXitFaxOptions* structure contains the values that you can predefined for the specific document.

Example 11-11 Structures specific to the fax options exit

```
*****  
/* FAX OPTIONS STRUCTURES */  
*****  
  
#define ARSCSXIT_FAX_RECIPIENT_ATTN_MAX 50  
#define ARSCSXIT_FAX_RECIPIENT_COMPANY_MAX 50  
#define ARSCSXIT_FAX_RECIPIENT_FAX_MAX 32  
#define ARSCSXIT_FAX_SENDER_FROM_MAX 50  
#define ARSCSXIT_FAX_SENDER_COMPANY_MAX 50  
#define ARSCSXIT_FAX_SENDER_PHONE_MAX 32  
#define ARSCSXIT_FAX_SENDER_FAX_MAX 32  
#define ARSCSXIT_FAX_SENDER_COVERPAGE_MAX 8  
  
typedef struct _ArsCSXitFaxOptions  
{  
    char recipient_attn[ARSCSXIT_FAX_RECIPIENT_ATTN_MAX + 1];  
    char recipient_company[ARSCSXIT_FAX_RECIPIENT_COMPANY_MAX + 1];  
    char recipient_fax[ARSCSXIT_FAX_RECIPIENT_FAX_MAX + 1];  
    char sender_name[ARSCSXIT_FAX_SENDER_FROM_MAX + 1];  
    char sender_company[ARSCSXIT_FAX_SENDER_COMPANY_MAX + 1];  
    char sender_phone[ARSCSXIT_FAX_SENDER_PHONE_MAX + 1];  
    char sender_fax[ARSCSXIT_FAX_SENDER_FAX_MAX + 1];
```

```
    char sender_coverpage[ARSCSXIT_FAX_SENDER_COVERPAGE_MAX + 1];
} ArsCSXitFaxOptions;
```

From the header file for fax options exit in Example 11-12, the input to the exit program is in the structure of *ArcCSXitApplGroup* and *ArcCSXitDocFields*, which corresponds to the application group information and the document fields. With this information, you can write a program and provide output using the structure of *ArsCSXitFaxOptions*. This allows you to customize the fax information, such as the recipient company and the fax number, based on the input such as the account number of the document. When a user faxes a document, it can be prefilled with the necessary recipient and fax number according to the document opened. Of course, the user is still free to modify the fax information options.

Example 11-12 Header file of the fax options exit

```
/*************************************************************************/
/* FAXOPTS - Fax Options Exit                                         */
/*                                                               */ 
/* This exit is for specialized applications and is not normally   */
/* used.                                                       */ 
/*                                                               */ 
/* INPUT: appl_grp                                              */
/*          doc_flds                                           */
/*                                                               */ 
/* OUTPUT: exitdata                                            */
/*                                                               */ 
/* RETURN_CODE:                                                 */
/*      0           -> Successful                            */
/*      Otherwise -> Failed                                */
/*                                                               */ 
/*************************************************************************/
int
ARSCSXIT_EXPORT
ARSCSXIT_API
FAXOPTS( ArcCSXitApplGroup *appl_grp,
          ArcCSXitDocFields *doc_flds,
          ArsCSXitFaxOptions *exitdata
);
```

Activating the fax options exit

To enable the fax options exit, place the compiled exit program **arsufax** in the bin/exits directory of the OnDemand installation root. For example, in AIX, place the program **arsufax** in the /usr/lpp/ars/bin/exits directory.

11.4.3 Load exit

The load exit is used to send notification after a document is loaded. The header file in Example 11-13 shows the information that can be used to incorporate into the notification message.

Example 11-13 Header file of the load exit

```
/*****************************************************************************  
/* LOADEXIT - Load Exit */  
/*  
/* To activate the load exit, the arsuload 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 */  
    unsigned long  deprecated; /* was bytes. Use report_bytes */  
    unsigned long  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 containg rows in non-UTF8 */  
    unsigned long  cp;        /* codepage file_l is in */  
    void          **hdl;       /* pointer to anchor for arsuload */  
    unsigned char  ColDelim;   /* Character used to delimit columns*/  
    ArcCSxitBigInt report_bytes; /* Number of bytes in report */  
} ArsCSxitLoadExit;  
  
int  
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, the application group name, and the size of the report. 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 **arsunload** in the bin/exits directory of OnDemand installation root. For example, in AIX, place the program **arsunload** in the /usr/lpp/ars/bin/exits directory.

11.4.4 Permission exit

The permission exit is more complex. It is used to customize permission in a more flexible way than the standard OnDemand administrative client can provide. This exit is called during login if the permission exit is turned on for folder and application groups. It is also called during a search when the permission exit is turned on for an SQL query string or document.

Example 11-14 shows the header file.

Example 11-14 Header file of permission exit

```
*****  
/* PERMEXIT - Permission Exit */  
/* To activate the permission exit on folder and application groups, */  
/* set the following variable in ars.ini: */  
/* SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1 in ars.ini */  
/* To activate the permission exit on documents, set the following */  
/* variable in ars.ini (please note that this exit can greatly */  
/* decrease OnDemand performance when performing a document query): */  
/* SRVR_FLAGS_DOCUMENT_EXIT=1 */  
/* To activate the permission exit on the sql query string, set the */  
/* following variable in ars.ini: */  
/* SRVR_FLAGS_SQL_QUERY_EXIT=1 in ars.ini */  
/* INPUT: userid */  
/* perm_exit */  
/* OUTPUT: */  
/* Check Folder */  
/* action == 1 (folder_perm) */  
/* Access to Folder using folder_name */
```

```

/*
 *access -> 0 no access
 */
/*
 *access -> 1 defaults to *PUBLIC access
 */
/*
 *access -> 2 grants access/primary folder
 */
/*
 *access -> 3 grants access/secondary folder
 */
/*
 */
/*
 Check Application Group
 */
action == 2 (appl_grp_perm)
/*
 Access to Application Group using appl_grp
 */
/*
 *access -> 0 no access
 */
/*
 *access -> 1 defaults to *PUBLIC access
 */
/*
 *access -> 2 grants access
 */
/*
 */
/*
 Check Document
 */
action == 3 (doc_perm)
/*
 Access to Document using appl_grp and doc
 */
/*
 *access -> 0 no access
 */
/*
 *access -> 1 grants access
 */
/*
 */
/*
 SQL Query String
 */
action == 4 (sql_query_perm)
/*
 Do not change the inp_sql or inp_sql_r values
 */
/*
 *access -> Is not used
 */
/*
 */
/*
 To Change the input SQL Query String
 */
/*
 Allocate and set the new string to out_sql
 */
/*
 Otherwise set out_sql = NULL and in_sql will be used
 */
/*
 */
/*
 To Change the input SQL Query Restriction String
 */
/*
 Allocate and set the new string to out_sql_r
 */
/*
 Otherwise set out_sql_r = NULL and in_sql_r will be used
 */
/*
 */
/*
 * RETURN_CODE:
 */
/*
 0 -> Successful
 */
/*
 Otherwise -> Failed
 */
/*
 *****/

```

```

typedef struct _ArcCSxitPermExit
{
/*
 * action
 * 1 - Folder
 * 2 - Application Group
 * 3 - Document
 * 4 - SQL Query String
 */
int action;
union
{

```

```

    struct
    {
        char *folder_name;
    } folder_perm;
    struct
    {
        ArcCSXitApplGroup appl_grp;
    } appl_grp_perm;
    struct
    {
        ArcCSXitApplGroup appl_grp;
        ArcCSXitDoc doc;
    } doc_perm;
    struct
    {
        ArcCSXitApplGroup appl_grp;
        char *in_sql;
        char *in_sql_r;
        char *out_sql;
        char *out_sql_r;
    } sql_query_perm;
} u;
} ArcCSXitPermExit;

int
ARSCSXIT_EXPORT
ARSCSXIT_API
PERMEXIT( char *userid, ArcCSXitPermExit *perm_exit, int *access );

```

The input of the exit program is the user ID and the information from the structure field *ArcCSXitPermExit*. The output is the access values of the different actions. The access values of the first two actions determine whether the user has the right to access the folder and application group during logon. The exit program can also change the SQL query and the SQL query restriction for the application group in action 4. Finally, the access value of action 3 determines the permission to retrieve the document into the hit list.

Example 11-15 shows a sample program flow.

Example 11-15 Sample program flow for the permission exit

```

Action 1
Check Folder permission using input from folder_perm
Based on your SQL code, output the user access permission
    If no access to folder
        return (access = 0)
    Elseif access defaults to *PUBLIC access
        return (access = 1)

```

```

Elseif grants access/primary folder
    return (access = 2)
Else grants access/secondary folder
    return (access = 3)

Action 2
Check Application Group permission using input from appl_grp_perm
Based on your SQL code, output the user access permission
    If no access to application group
        return (access = 0)
    Elseif access defaults to *PUBLIC access
        return (access = 1)
    Elseif grants access
        return (access = 2)

Action 4
Check the SQL Query String using input from sql_query_perm before searching
starts. Based on your SQL code, output the new SQL query string if needed
User will use the new sql string to perform query if it is available
    If change in SQL query string is needed
        set out_sql = new query string
    Else if no change is needed (in_sql will be used)
        set out_sql = null
    If change in SQL query restriction string is needed
        set out_sql_r = new query string
    Else if no change is needed (in_sql_r will be used)
        set out_sql_r = null
        return (access = not use)

Action 3
Check the document access permission after using the SQL query to search
using the input from doc_perm and based on your SQL code
    If no access to document (document will not be shown on hitlist)
        return (access = 0)
    Else grants access
        return (access = 1)

```

The output of the program is in a structure of ArcCSXitPermExit, with the final access values and SQL queries. The permission exit overrides the permission defined on the OnDemand administrative client. It can be used for such occasions as when you want specific users or groups to view certain financial reports only during a certain time of the year, but you do not want to change the permission from the administrative client.

Activating the permission exit

The permission exit can be activated by specifying the respective variables in the ARS.INI file with the `arsuperm` exit program placed in the bin/exits directory of the OnDemand installation root. The ARS.INI file is found in the config directory of the OnDemand installation root.

For AIX, the ARS.INI file to be modified is in the `/usr/lpp/ars/config/ars.ini` directory. For the exit program, it should be placed in the `/usr/lpp/ars/bin/exits/arsuperm` directory.

You set the following variables to activate the different permissions in the ARS.INI file:

- ▶ To activate the folder or the application permission

`SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1`

- ▶ To activate the SQL query exit

`SRVR_FLAGS_SQL_QUERY_EXIT=1`

- ▶ To activate the document permission exit

`SRVR_FLAGS_DOCUMENT_EXIT=1`

11.4.5 Client retrieval preview exit

The client retrieval preview user exit allows for the modification of document data prior to the data being retrieved from the server. It is called during 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:

- ▶ Remove pages from the document, for example, banner pages, title pages, or all pages except the summary page.
- ▶ Remove specific words, columns of data, or other information from the document. That is, omit (“white out”) sensitive information such as salaries, social security numbers, and birth dates.
- ▶ Add information to the document, for example, a summary page, data analysis information, and Confidential or Copy statements.
- ▶ Reformat data contained in the document. For example, reorder the columns of data.

The client retrieval user exit point might be enabled for more than one application. However, all applications must be processed by the same user-written program (only one user-written program is supported). The system passes the name of the application that is associated with the document to the user-written program. The user-written program can perform processing based on the application, or it can perform the same processing for all documents regardless 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 11-16 shows the header file of the client retrieval preview exit.

Example 11-16 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                                  */
/*      pAppl                                    */
/*      pDoc                                      */
/*
 */
/* OUTPUT:                                         */
/*
 */
/* RETURN_CODE:                                     */
/*      0            -> Successful                */
/*      Otherwise -> Failed                     */
/*
 */
/********************************************/
typedef struct _ArsCSxitPrepExit
{
    char*           pUserId;        /* Logged on userid */
    char*           pInFileName;    /* File name for document data */

    /* File name for modified data */
    char           OutFileName[ARCCSXIT_PATH_MAX + 1];

    char*           pUserParms;    /* User defined parms from appl */
    ArcCSxitApplGroup* pApplGrp;   /* Appl Grp info */
```

```
ArcCSXitAppl*      pAppl;          /* Application info */
ArcCSXitDoc*       pDoc;           /* Doc handle, field info */
} ArsCSXitPrepExit;

int
ARSCSXIT_EXPORT
ARSCSXIT_API
PREPEXIT( ArsCSXitPrepExit* prep );
```

For example, you can program 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 exit

To activate the client retrieval 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 OnDemand installation root. When the option is selected, the user-written program is called any time a request is made to retrieve a document.

Any information that is specified in the Parameters field is passed to the user-written program. For example, in AIX, place the **arsuprep** program in the /usr/lpp/ars/bin/exits directory.

The retrieval preview user exit might be enabled for all data types, except for None.

For more information, refer to *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232.

11.4.6 Security exit

The OnDemand security exit is available for all platforms. By default, the iSeries server activates the security exit and uses OS/400 security. If the security exit is not enabled, then the OnDemand user ID and password have no relation to the OS/400 user ID and password and all the OnDemand System parameter settings are honored. Enabling or disabling this exit can be done at an individual instance level.

The security exit uses a specific structure called *ArcCSXitSecurityAction* and *ArcCSXitSecurityRC* as shown in Example 11-17.

Example 11-17 Structures specific to the security exit

```
*****  
/* SECURITY STRUCTURES */  
*****  
typedef enum _ArcCSXitSecurityAction  
{  
    ARCCSXIT_SECURITY_USER_LOGIN,  
    ARCCSXIT_SECURITY_USER_ADD,  
    ARCCSXIT_SECURITY_USER_DELETE,  
    ARCCSXIT_SECURITY_USER_UPDATE  
} ArcCSXitSecurityAction;  
  
typedef enum _ArcCSXitSecurityRC  
{  
    ARCCSXIT_SECURITY_RC_OKAY,  
    ARCCSXIT_SECURITY_RC_PERMS,  
    ARCCSXIT_SECURITY_RC_PASSWD_CHNG,  
    ARCCSXIT_SECURITY_RC_FAILED,  
    ARCCSXIT_SECURITY_RC_OKAY_BUT_VALIDATE_IN_OD  
} ArcCSXitSecurityRC;
```

These structures are used in the security exits as demonstrated in Example 11-18, the header file of the security exit. Notice that there is a new addition to the header file, the parameter *clnt_id*, the client identifier. This identifier contains the host name and the IP address of the user who is accessing the server. This information can be used to further classify users and to grant access based on predefined criteria.

Example 11-18 Header file of a security exit

```
*****  
/* SECURITY - Security Exit */  
/*  
/* To activate the security exit, set the following variable in the */  
/* appropriate OnDemand instance section in the ars.ini file: */  
/*  
/*     SRVR_FLAGS_SECURITY_EXIT=1 */  
/*  
/* 1) User Login */  
/*     On Input:  action == ARCCSXIT_SECURITY_USER_LOGIN */  
/*  
/*     INPUT: cur_userid - Userid */  
/*             cur_passwd - Password */  
/*  
/*             clnt_id - Client's hostname and IP address */
```

```

/*      OUTPUT:                                */
/*
/* 2) User Add                                */
/*      On Input:  action == ARCCSXIT_SECURITY_USER_ADD   */
/*
/*      INPUT: act_userid - Actual User doing the add    */
/*              new_userid - Userid to add                */
/*              new_passwd - Password                   */
/*              clnt_id - Client's hostname and IP address */
/*
/*      OUTPUT:                                */
/*
/* 3) User Delete                             */
/*      On Input:  action == ARCCSXIT_SECURITY_USER_DELETE */
/*
/*      INPUT: act_userid - Actual User doing the delete */
/*              cur_userid - Userid to delete            */
/*              clnt_id - Client's hostname and IP address */
/*
/*      OUTPUT:                                */
/*
/* 4) User Update                            */
/*      On Input:  action == ARCCSXIT_SECURITY_USER_UPDATE */
/*
/*      INPUT: act_userid - Actual User doing the update */
/*              cur_userid - Current userid             */
/*              cur_passwd - Current password          */
/*              new_userid - New userid                 */
/*              new_passwd - New Password              */
/*                      If NULL, then no password change */
/*              clnt_id - Client's hostname and IP address */
/*
/*      OUTPUT:                                */
/*
/* RETURN_CODE:                                */
/*      ARCCSXIT_SECURITY_RC_OKAY      -> Successful   */
/*      ARCCSXIT_SECURITY_RC_PERMS     -> No permission. */
/*      ARCCSXIT_SECURITY_RC_PASSWD_CHNG -> Only valid on Login. */
/*                                         Login is successful, */
/*                                         password must be */
/*                                         changed.           */
/*      ARCCSXIT_SECURITY_RC_FAILED    -> Failed        */
/*
/* NOTES: (output) msg is ARCCSXIT_MAX_SRVR_MESSAGE_SIZE bytes in */
/* size.  If the return code != ARCCSXIT_SECURITY_RC_OKAY       */
/* then if msg[0] != '\0', then this will be the message        */
/* displayed by the client.  Otherwise the client will use      */
/* its native message text.                                     */
*/

```

```

/*
 *      A new parameter, clnt_id, has been added to allow for      */
/*      checking the client's hostname and IP addressss. It is      */
/*      passed as a character string with the format of      */
/*      "hostname.domainname IPaddress". An Example would be      */
/*      "client.some.company.com 100.100.100.1"      */
/*
 ****
ArcCSxitSecurityRC
ARSCSXIT_EXPORT
ARSCSXIT_API
SECURITY( char *act_userid,
          char *cur_userid,
          char *cur_passwd,
          char *new_userid,
          char *new_passwd,
          ArcCSxitSecurityAction action,
          char *msg,
          char *clnt_id
);

```

11.4.7 Security exit on z/OS

On a z/OS server, nothing goes without operating security that is usually provided by Security Access Facility (SAF) of the operation system. The security exit is not needed if you want to run your OnDemand Server with the internal OnDemand security only. The security exit is implemented to allow the communication with an external security manager such as RACF. The OnDemand security system interface exit allows an installation to control the following events or activities:

- ▶ Logging on
- ▶ Changing a password
- ▶ Adding a user ID or deleting user ID by using the OnDemand administrative functions
- ▶ Obtaining access to an OnDemand folder
- ▶ Obtaining access to an OnDemand application group

If any of the events or activities occurs, a user-written exit routine can interact with a security system, such as RACF, to determine whether the given activity is allowed.

Also, the ARCCSXIT_SECURITY_OKAY_BUT_VALIDATE_IN_OD return code option allows a user to perform some action on a request and then allows OnDemand to perform the standard security processing. An example of this is to not allow a “new” password to match an “old” password in a change-password request; the password must be changed.

Table 11-2 lists the modules or executables that are shipped with OnDemand.

Table 11-2 Security exit modules

Module	Description
ARSUPERM	This c-module provides the interface between the OnDemand system and the ARSUSECX module.
ARSUSEC	This c-module provides the interface between the OnDemand system and the ARSUSECX module.
ARSUSECA	Mapping of the data structure presented to the exit routine is associated with the exit point defined by ARSUSEC in Assembler.
ARSUSECH	Mapping of the data structure presented to the exit routine is associated with the exit point defined by ARSUSEC in C.
ARSUSECJ	This is a sample JCL stream for assemble and bind ARSUSECX and ARSUSECZ.
ARSUSECX	This is the interface module for the MVS Dynamic Exit Facility.
ARSUSECZ	This is the Security Exit Module Sample.

Note: The security exit is an enhancement that is not shipped with the basic code. It is available with PTF UQ58458 UQ59190.

All modules are found in the SARSINST library after applying the PTF. 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 a CICS® environment. The kernel code was updated to allow external security. The OnDemand Kernel code calls a dynamic link library (DLL) as an interface to the exit. Modules ARSUSEC and ARSUPERM, provided as C source code modules and as executables, fulfill this function. There is no need to change and recompile them.

The source is delivered mainly for understanding the entire security system exit. If you want to change them, they have to be recompiled and bound as a C 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 with the PTF. It shows how to perform security checks with a Security Exit Facility (SAF) interface. RACF is a program that uses SAF. The ARSUSECH is a C source code module that passes the data structure as input for every exit (ARSUSECZ) that is provided. The ARSUSEA provides the same in assembler language.

Note: You can have more than one security exit defined to the MVS Dynamic Exit Facility. For example, 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.

Security systems other than SAF

The sample provided by IBM is an SAF sample. However, there are installations that use their own security system or use it as an enhancement together with SAF environment. These systems can be accessed if they provide a proper 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 Security Exit Facility interface, this part must be updated to call this external security system. For demonstration purposes, Example 11-19 shows the access to an application group code sample. This sample issues the RACROUTE macro. If a different external security manager is used, this code must be updated for a proper call of this system.

Example 11-19 Sample for an application group request

```
TITLE 'HACAPGP: Process an Application Group Access Request'
**** HACAPGP: Process an Application Group Access Request
*
*
* Function:
*      This procedure processes a request for Read access to an
*      OnDemand Application Group.
*
*
* Inputs:
*      Registers:
*          R10: Points to the WORKMAP structure.
```

```
*           R11: Points to the ARSUSECA structure.  
*  
*  
* Outputs (normal):  
*     Registers:  
*         R0: Unchanged.  
*         R1: Unchanged.  
*  
*         R15: Contains one of the following return  
*                 code values (see the ARSUSECA DSECT  
*                 for return code details):  
*  
*                 ARSUSECA_RCNORM  
*                 ARSUSECA_RCPERMS  
*  
*  
* Outputs (error):  
*     None.  
*  
*  
* Exits (normal):  
*     Via Program Return (PR)  
*  
*  
* Exits (error):  
*     None.  
*  
*  
* Linkage:  
*     Via the ICALL macro interface.  
*  
*  
* Special Considerations:  
*     None.  
*  
*  
* Algorithm:  
*     . The requesting User ID and the target Application Group  
*           Name strings are copied to the local work area.  
*  
*           Note that:  
*               . These strings will be truncated if they are longer  
*                   than the maximum length as defined by the ARSUSECA  
*                   DSECT.  
*  
*               . It is expected that the SAF conforming security  
*                   system will enforce the length and character set  
*                   restrictions associated with User ID and resource  
*                   profile name strings.
```

```

*
*. A RACROUTE AUTH request is issued to determine if the
* requesting User ID is to be granted Read access to the
* Application Group.
*
*. The procedure return code is set to ARSUSECA_RCNORM for
* the following situations:
*
*- The SAF conforming security system has granted
* access.
*
*- The SAF conforming security system has not made a
* decision. This can occur, for example, when the
* Resource Class is not defined to the security
* system or when no profile exists for the named
* entity.
*
Otherwise, the procedure return code is set to
ARSUSECA_RCPERMS.
*
*. Exit
*
* -----
*      SPACE 2
HACAPGP DC OH'0'
PUSH USING
EJECT ,
* -----
*
* Copy the requesting User ID string, blank padding or truncating
* as required.
*
* -----
*      SPACE 2
*      L     R14,ARSUSECA_CURUIDP   Fetch the ptr to the
*                                         ID string of the user
*                                         being updated.
*
*      L     R15,ARSUSECA_CURUIDL   Fetch the User ID string
*                                         length.
*
*      LA    R0,WKUIDS           Set the MVCL 'To' adrs.
*      LA    R1,L'WKUIDS(,0)       Set the MVCL 'To' len.
*
*      STC   R15,WKUIDL          Set the User ID length
*                                         field as required by
*                                         RACROUTE.
*

```

```

        CLR    R15,R1           Is the User ID string
*                                         too long to be contained
*                                         in the copy area --
*
        BNH    HAAG200          Br if not
        STC    R1,WKUIDL        Else truncate the string.
*
        HAAG200 DC    OH'0'
        ICM    R15,B'1000',BLANKS Set the MVCL pad value.
*
        MVCL   R0,R14           Copy the User ID string
*                                         to the local work area.
        EJECT ,
*
* -----
*
* Copy the Application Group name (i.e., the name of the entity
* to which access is being requested), blank padding or
* truncating as required.
*
* The actual entity name string (i.e., exclusive of the trailing
* blanks) is translated to convert any embedded blanks or invalid
* characters to the character value defined by the RPNINV equate.
* In addition, lower case characters *MAY* be converted to upper
* case.
*
* -----
        SPACE 2
        L     R14,ARSUSECA_AGNMP Fetch the ptr to the
*                                         Application Group name.
*
        L     R15,ARSUSECA_AGNML Fetch the Application
*                                         Group name length.
*
        LA    R0,WKENTNM         Set the MVCL 'To' adrs.
        LA    R1,L'WKENTNM(,0)   Set the MVCL 'To' len.
*
        ICM    R15,B'1000',BLANKS Set the MVCL pad value.
*
        MVCL   R0,R14           Copy the Application Group
*                                         name to the local work
*                                         area.
*
        L     R15,ARSUSECA_AGNML Fetch the Application
*                                         Group name length.
*
        LA    R14,L'WKENTNM(,0) Load the length of the
*                                         entity name buffer area.
*
        CLR    R15,R14           * Br if the name string

```

```

BNH    HAAG400          * was not truncated.
*
LR     R15,R14           Else use the truncated len.
HAAG400 DC    OH'0'
BCTR   R15,0              * Convert the string to
EX     R15,TRENTNM       * valid characters.
*
SLL    R14,16             * Set the entity buffer
ST     R14,WKENTBFL      * length fields.
EJECT ,
*
* -----
*
* Issue a RACROUTE AUTH request to determine if the user has Read
* access to the Application Group.
*
* -----
SPACE 2
MVI    WKSACFCLL,L'WKSACFCLN  * Build the SAF Resource
MVC    WKSACFCLN,ARSAGRN     * Class Name area.
*
MVC    WKRACFPL(LNRACAUT),SKRACAUT
*                                         Build a RACROUTE AUTH
*                                         template plist.
*
XR    R15,R15             * Zero the ACEE pointer
ST    R15,WKACEEP         * return area.
SPACE 2
RACROUTE REQUEST=AUTH,          Validate access authority      +
CLASS=WKSACFCLS,               SAF Resource Class area      +
ATTR=READ,                     Authority requested        +
ENTITYX=(WKENTBUF,NONE),       Resource Profile Name area  +
USERID=WKUIDS,                 User ID to validate       +
WORKA=WKRACWKA,                SAF work area            +
RELEASE=2608,                  OS/390 2.8 level        +
MSGTRRN=NO,                    Do not return messages  +
MF=(E,WKRACFPL)
EJECT ,
*
* -----
*
* The RACROUTE operation is considered successful if the
* validation was completely successful or the security system
* made no decision.
*
* -----
SPACE 2
LA     R14,WKRACFPL       Set the ptr to the
*                                         RACROUTE interface
*                                         plist.

```

```

        USING SAFP,R14
*
*           LA    R2,ARSUSECA_RCNORM(,0) Assume the RACROUTE
*           operation completed
*           successfully.
*
*           LTR   R15,R15          Is this the case --
*           BZ    HAAGFIN          Br if so.
*
*           CL    R15,SAFRNOD      Was no decision made by
*                           the security system --
*
*           BE    HAAGFIN          Br if so.
*
*           LA    R2,ARSUSECA_RCPERMS(,0) Else indicate the user
*                           is not to be granted
*                           access to the Application
*                           Group.
*
*           DROP  R14          SAFP (ICHSAFP) base
*           EJECT ,
* -----
*
*           * Delete the newly created ACEE (if it exists) and exit.
*
*           * At entry to this code segment it is expected that GPRs are
*           loaded as follows:
*
*           *     R2: Contains the procedure return code value.
*
* -----
*
*           SPACE 2
HAAGFIN DC    OH'0'
*           L    R1,WKACEEP       Fetch the potential ptr
*                           to the newly constructed
*                           ACEE.
*
*           SPACE 2
*           ICALL DELACEE       Delete the new ACEE.
*           SPACE 2
HAAGXIT DC    OH'0'
*           LR   R15,R2          Set the procedure
*                           return code.
*
*           EREG  R0,R1          Restore the entry regs
*           PR   ,                And exit
*
*           POP   USING

```

OnDemand SAF resource classes

You must define SAF resource classes ARS1FLDR and ARS1APGP for the folders and application group. Refer to Appendix E in *IBM Content Manager OnDemand for z/OS and OS/390 - Configuration Guide*, GC27-1373, for more information about the resource classes.

Important: Even if the security exit can check the UID and password against SAF or other security systems, every user must be defined in OnDemand in every instance. You can use the ARSADM program to create users in batch mode, as a command from the UNIX System Services command line and using a file as input.

Activating the security exit

Activation of the security exit is controlled by settings in the ARS.INI file, which resides in the /usr/lpp/ars/config directory for AIX.

You can enable the exit for the following events:

- ▶ Logon
- ▶ Changing the password
- ▶ Adding or deleting a user ID via the OnDemand administrator interface

To enable the exit for these events, you must add the following statement to the ARS.INI file:

```
SRVR_FLAGS_SECURITY_EXIT=1
```

For activation of the application group and folder permission exit, refer to 11.4.4, "Permission exit" on page 367.

Activating the security exit in a z/OS environment

The module ARSUSECX interfaces with the MVS Dynamic Exit Facility to:

- ▶ Define the logical exit point name, ARS.SECURITY
- ▶ Route the control to a set of associated exit routines and process the results of their execution

Note: The sample is designed to process 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, which is done by associating a logical exit point (ARS.SECURITY). In this example, the MVS Dynamic Exit Facility provides several methods performing this association. You can use the PROGXX statement in Sys1.Parmlib to define exits to the Dynamic Exit Facility at IPL time (Example 11-20).

Example 11-20 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 found in LPA or a LNLKLST data set.

The security exit can only handle the functions that we described earlier. If you want to restrict access to folder and application groups based on index values, you can do this with the internal OnDemand security. The restriction for an application group is maintained by RACF. When a user has access to the application group, there is no way to limit the access to this application group with any external security. To limit access to specific application group data, enter a Query Restriction to the Application Group to create an SQL “where clause”.

Figure 11-3 shows a query that is restricted to statements with balance exceeding 200. This query restriction is for all users of the system (*PUBLIC) that do not have a separate query restriction.

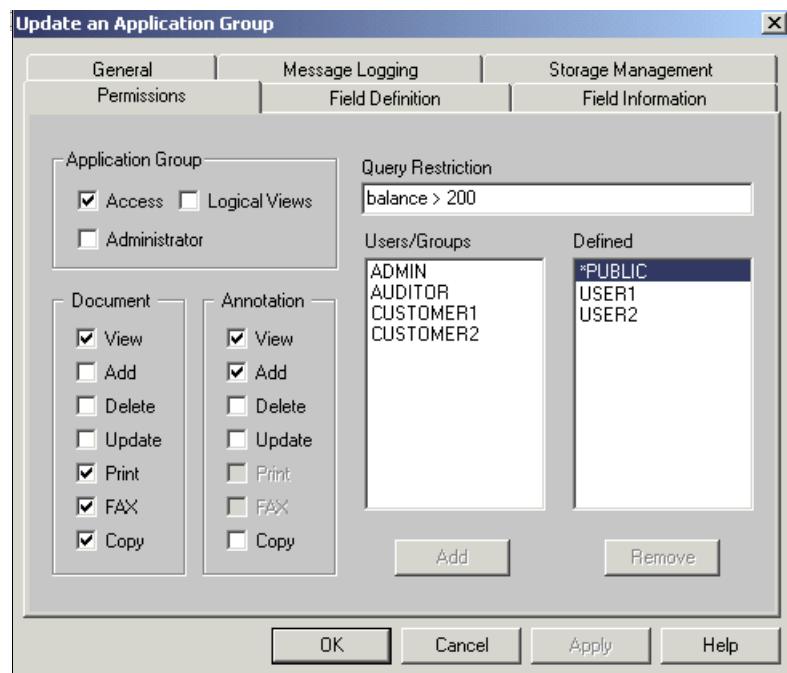


Figure 11-3 Setting the query restriction

11.4.8 ARSYSPIN and sample APKACIF exit on z/OS

The JES Spool Capture facility ARSYSPIN and the sample APKACIF exit on z/OS are provided by PTF PQ57769. ARSYSPIN provides a means to collect and consolidate the JES spool (SYSOUT) data set into one or more files so they can be archived by OnDemand. The facility executes as a started task in its own address space. A control statement file is used to provide ARSYSPIN parameters. These parameters specify JES Spool file selection criteria (for example, the sysout class taken for capture output) and other operational characteristics.

ARSYSPIN 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 OnDemand using the ARSLOAD program. ARSYSPIN invokes ARSLOAD when sufficient data has been 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.

Special considerations for APKACIF exits written in COBOL

The provided sample exit is written as a COBOL main program. To prevent the language environment from creating and destroying the COBOL runtime environment, each time the ARSSPVIN is called. A CEEUOPT CSECT must be assembled and link-edited with the COBOL object code. You must update the sample taken from CEE.SCEESAMP and specify the following option:

RTEREUS=(ON)

In addition, you must be sure that the resulting module is link-edited as NOT RE-ENTRANT and NOT REUSEABLE. This is required to allow the local variables within the COBOL exit code to retain their values. This exit is invoked several times during an ACIF run. See Example 11-21, the JCL sample for details. The sample source code can be found in the SARSINST library member ARSSPVIN.

Example 11-21 JCL sample

```
//ALLOC EXEC PGM=IEFBR14
//OBJ      DD DSN=&&OBJ,DISP=(NEW,PASS),
//           UNIT=VIO,SPACE=(TRK,(30,,5)),
//           DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB,DSORG=P0)
///*
///*
//COBOL    EXEC PGM=IGYCRCTL,REGION=0M,
//           PARM=('NODYNAM,LIB,LIST,MAP,OBJECT',
//                  'RENT,APOST,TRUNC(BIN),NOSEQ,XREF')
//STEPLIB   DD DISP=SHR,DSN=COBOL.V2R1MO.SIGYCOMP
//SYSPRINT  DD SYSOUT=*
//SYSLIB    DD DISP=SHR,DSN=ARSV710.0DMP710.SARSINST
//           DD DISP=SHR,DSN=CEE.SCEESAMP
//SYSLIN    DD DISP=(OLD,PASS),DSN=&&OBJ(ARSSPVIN)
//SYSUT1    DD UNIT=VIO,SPACE=(460,(3500,100))
//SYSUT2    DD UNIT=VIO,SPACE=(460,(3500,100))
//SYSUT3    DD UNIT=VIO,SPACE=(460,(3500,100))
//SYSUT4    DD UNIT=VIO,SPACE=(460,(3500,100))
//SYSUT5    DD UNIT=VIO,SPACE=(460,(3500,100))
//SYSUT6    DD UNIT=VIO,SPACE=(460,(3500,100))
//SYSUT7    DD UNIT=VIO,SPACE=(900,(7000,100))
```

```

//SYSIN      DD DISP=SHR,DSN=ARSV710.ODMP710.SARSINST(ARSSPVIN)
//*
//* -----
//*
//ASM      EXEC PGM=ASMA90,REGION=0M,
//          PARM=(NOBJECT,DECK,NOTERM,XREF(SHORT),LIST(MAX),
//                 ASA,MXREF(FULL))
//STEPLIB   DD DISP=SHR,DSN=HLASM.V1R4MO.SASMMOD1
//          DD DISP=SHR,DSN=HLASM.V1R4MO.SASMMOD2
//SYSPRINT  DD SYSOUT=*
//SYSTERM   DD DUMMY
//SYSADATA  DD DUMMY
//SYSLIN    DD DUMMY
//SYSPUNCH  DD DISP=(OLD,PASS),DSN=&&OBJ(CEEUOPT)
//SYSUT1    DD UNIT=SYSALDDA,SPACE=(CYL,(10))
//SYSLIB    DD DISP=SHR,DSN=CEE.SCEEMAC
//          DD DISP=SHR,DSN=CEE.SCEESAMP
//          DD DISP=SHR,DSN=SYS1.MACLIB
//          DD DISP=SHR,DSN=SYS1.MODGEN
//SYSIN     DD *
******
*/
*/
/* OS/390 2.9 LANGUAGE ENVIRONMENT */
/* LICENSED MATERIALS - PROPERTY OF IBM. */
/* 5647-A01 5688-198 */
/* (C) COPYRIGHT IBM CORP. 1991, 2000 */
/* ALL RIGHTS RESERVED */
/* US GOVERNMENT USERS RESTRICTED RIGHTS - USE, */
/* DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP */
/* SCHEDULE CONTRACT WITH IBM CORP. */
/* STATUS = HLE6609 */
*/
CEEUOPT CSECT
CEEUOPT AMODE ANY
CEEUOPT RMODE ANY
        CEEXOPT ABPERC=(NONE), +  

          ABTERMENC=(ABEND), +  

          AIXBLD=(OFF), +  

          ALL31=(OFF), +  

          ANYHEAP=(16K,8K,ANYWHERE,FREE), +  

          BELOWHEAP=(8K,4K,FREE), +  

          CBLOPTS=(ON), +  

          CBLPSHPOP=(ON), +

```

CBLQDA=(OFF),	+	
CHECK=(ON),	+	
COUNTRY=(US),	+	
DEBUG=(OFF),	+	
DEPTHCONDLMT=(10),	+	
ENVAR=''),	+	
ERRCOUNT=(0),	+	
ERRUNIT=(6),	+	
FILEHIST=(ON),	+	
HEAP=(32K,32K,ANYWHERE,KEEP,8K,4K),	+	
HEAPCHK=(OFF,1,0),	+	
HEAPPOLLS=(OFF,8,10,32,10,128,10,256,10,1024,10,2048,	+	
10),	+	
INFOMSGFILTER=(OFF,,,),	+	
INQPCOPN=(ON),	+	
INTERRUPT=(OFF),	+	
LIBRARY=(SYSCEE),	+	
LIBSTACK=(4K,4K,FREE),	+	
MSGFILE=(SYSOUT,FBA,121,0,NOENQ),	+	
MSGQ=(15),	+	
NATLANG=(ENU),	+	
NOAUTOTASK=,	+	
NONONIPTSTACK=(4K,4K,BELOW,KEEP),	+	
NOTEST=(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	+
RTLS=(OFF),	+	
SIMVRD=(OFF),	+	
STACK=(128K,128K,BELOW,KEEP),	+	
STORAGE=(NONE,NONE,NONE,8K),	+	
TERMTHDACT=(TRACE,,96),	+	
THREADHEAP=(4K,4K,ANYWHERE,KEEP),	+	
TRACE=(OFF,4K,DUMP,LE=0),	+	
TRAP=(ON,SPIE),	+	
UPSI=(00000000),	+	
VCTRSAVE=(OFF),	+	
VERSION=''),	+	
XUFLOW=(AUTO)	+	

```
      END  ,
/*
/*
/** -----
/** -----
//LKED      EXEC PGM=IEWL,COND=(4,LT,COBOL),
//          PARM='CASE=MIXED,COMPAT=CURR,LIST,LET,MAP,XREF'
//SYSPRINT DD SYSOUT=*
//SYSUT1    DD UNIT=VIO,SPACE=(TRK,(30)),DSN=&&LUT1
//SYSLIB    DD DISP=SHR,DSN=CEE.SCEELKED
//OBJECT    DD DISP=(OLD,DELETE),DSN=&&OBJ
//SYSLMOD   DD DISP=SHR,DSN=RAICER.SAMPLE.LOADLIB
//SYSLIN    DD *
           INCLUDE OBJECT(ARSSPVIN)
           INCLUDE OBJECT(CEEUOPT)
           ENTRY ARSSPVIN
           NAME ARSSPVIN(R)
/*

```

Enabling the exit

To activate the exit, you must add the executable into a loadlib in Steplib (ARSLOAD) procedure. You must also supply the ACIF control statement Inputexit = ARSSPVIN. You can do this when you update an application in the Indexer Properties window (Figure 11-4). When getting to the Indexer Information, you can modify them by typing in the statement with a keyboard or using sample data and specifying the exit in the Exit panel.

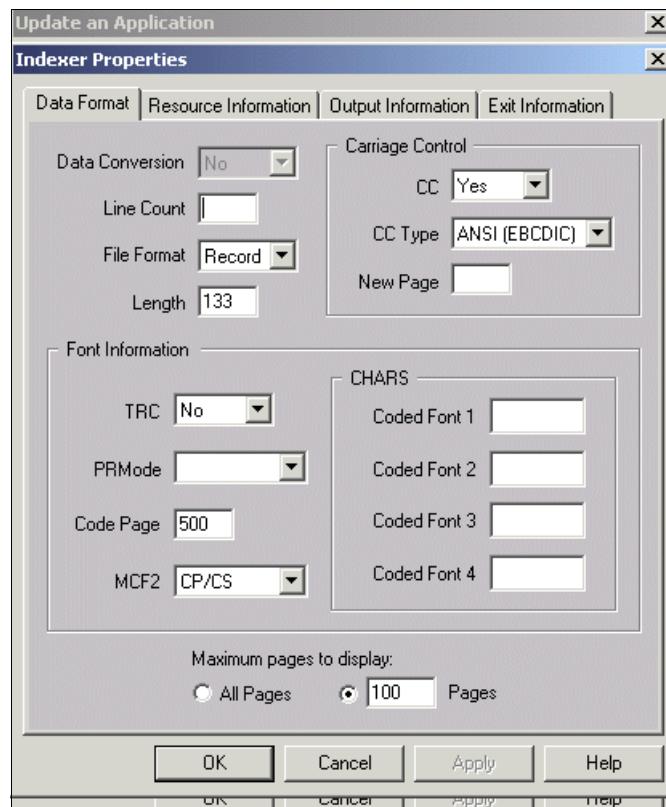


Figure 11-4 Indexer Properties window

Click the **Exit Information** tab (Figure 11-5), and in Input Records, type the name of the exit. Then click **OK**.

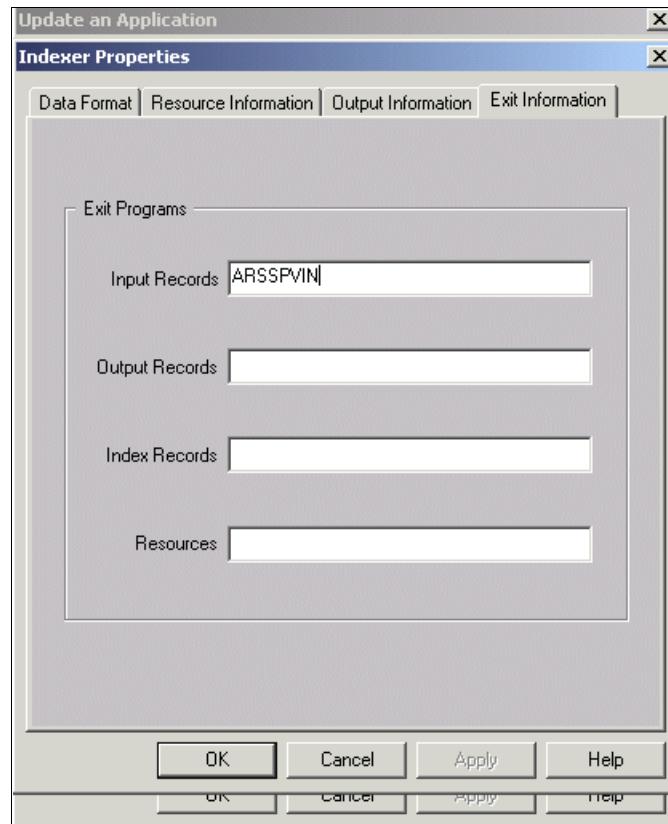


Figure 11-5 Specify the exit load module name

Specifying this information updates the indexer information as shown in Example 11-22.

Example 11-22 Index information

```
FIELD3=0,105,8,(TRIGGER=1,BASE=0)
FIELD4=-1,77,13,(TRIGGER=1,BASE=0)
INDEX1=X'998587899695',FIELD1,(TYPE=GROUP,BREAK=YES)      /* region      */
INDEX2=X'9985979699A395819485',FIELD2,(TYPE=GROUP,BREAK=YES) /* reportname   */
INDEX3=X'998481A385',FIELD3,(TYPE=GROUP,BREAK=YES)        /* rdate       */
INDEX4=X'99858789969595819485',FIELD4,(TYPE=GROUP,BREAK=YES) /* regionname   */
DCFPAGENAMES=NO
UNIQUEBNGS=YES
IMAGEOUT=ASIS
INDEXOBJ=GROUP
```

```
INDEXSTARTBY=1
INSERTIMM=NO
RESTYPE=NONE
INPEXIT=ARSSPVIN <----- UPDATE !
```

Note: If you are running OnDemand on z/OS, the ACIF indexer is running in an OS/390 environment. The normally provided Parmfile in JCL for ACIF is now provided as the indexer information in the application definition.

11.4.9 Storage management external cache exit

The storage management external cache exit is used to retrieve data from external storage. Depending on your programs, the external cache can be just a file from a directory or you can interface with other software to retrieve documents from other applications.

From the header file as shown in Example 11-23, the input is the application group information, ArcCSxitApplGroup, and the document information, ArcCSxitDoc. The output is the data if it is available.

Example 11-23 Header file for storage management external cache exit

```
*****
/* SMEXTCAC - Storage Management External Cache Exit */
/*
/* This exit is invoked only when data is to be retrieved from an */
/* Application Group that is defined with the External Cache setting */
/* checked. This exit is for specialized applications and is not */
/* normally used. */
/*
/* 1) Don't return data, only validate whether the document exists. */
/*   On Input: buf == NULL */
/*
/* INPUT: appl_grp */
/* doc */
/* buf */
/*
/* OUTPUT:
/*   *buf_len = 0 -> Data is not in external cache
/*           -> Otherwise data is in external cache */
/*
/* 2) Return document data.
/*   On Input: buf != NULL */
/*
/* INPUT: appl_grp */
/* doc */
/*   *buf_len -> #of bytes to retrieve */
```

```

/*
 *      OUTPUT:
 *          *buf    -> Allocated memory and document data.
 *                  Memory is freed by OnDemand using the
 *                  free() function
 *          *buf_len -> Length of *buf (should be same as on input)
 *                  -> If 0, then data does not exist within
 *                  external cache.
 *
 *      RETURN_CODE:
 *          0        -> Successful
 *          Otherwise -> Failed
 */
/*************************************************************************/
int
ARSCSXIT_EXPORT
ARSCSXIT_API
SMEXTCAC( ArcCSxitApplGroup *appl_grp,
           ArcCSxitDoc *doc,
           char **buf,
           unsigned long *buf_len
);

```

To use this exit, you must first load the index of the documents into OnDemand, and select **External Cache** when the application group is created. When the user retrieves the document from OnDemand based on the indexes, the exit is activated to pull the document from respective location.

Activating the storage management external cache exit

The exit is only activated when a user retrieves a document data that is stored in external cache. The **smextcac** exit program should be placed in the bin/exits directory of the OnDemand installation root.

11.4.10 Tablespace creation exit

The OnDemand tablespace creation exit allows an installation to take action when OnDemand creates a tablespace, table, or index tables that will be used to store application index data. The exit is not called for the OnDemand system tables. The tablespace creation exit is used to modify the way OnDemand creates tablespaces, tables or indexes. For table and index creation, the installation can alter the SQL that will be used to create the table or index.

You can also use this exit to perform other actions during a tablespace creation. This is useful if you must change default parameters for the tablespace, the table, or the indexes. The changes only affect new creations.

Example 11-24 shows the header file for the tablespace creation exit.

Example 11-24 Header file for the tablespace create exit

```
*****  
/* TBLSPCRT - Tablespace Create Exit */  
/*  
/* To activate the tablespace 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 */  
/*create_table_sql */  
/*action */  
/*  
/* OUTPUT:  
/*  
/* 1) OnDemand will invoke the exit with action == 1 */  
/* so that the exit can create the tablespace (tblsp_name) */  
/* *created -> 0 exit did not create the tablespace, */  
/* OnDemand needs to create the tablespace */  
/* *created -> 1 exit created the tablespace */  
/*  
/* 2) OnDemand will then invoke the exit with action == 2 */  
/* so that the exit can create the table (table_name) */  
/* inside of the tablespace (tblsp_name) using */  
/* (sql) */  
/* *created -> 0 exit did not create the table, */  
/* OnDemand needs to create the table */  
/* *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 tablespace (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 */  
/* *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 */  
/*
```

```
/* RETURN_CODE:  
/*      0          -> Successful  
/*      Otherwise -> Failed  
/*  
*****
```

You can use SQL code to customize the following actions:

- ▶ Creating a tablespace
- ▶ Creating a table
- ▶ Creating an index
- ▶ Other additional action

If you do not customize the action, OnDemand uses the defaults. Example 11-25 shows a sample program flow.

Example 11-25 Sample program flow

```
Action 1  
Is there a need to customize the creation of the tablespace?  
  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 tablespace creation exit

The exit is turned on by setting the following parameter in the ARS.CFG file, which is located in the config directory of the 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=absoulte path name
```

For AIX, it is in the /usr/lpp/ars/config/ars.cfg file, and the variable to be set is as follows:

```
ARS_DB_TABLESPACE_USEREXIT=/usr/lpp/ars/bin/exits/arsutbl
```

For this example, you must place the **arsutbl** exit program in the /usr/lpp/ars/bin/exits directory of the OnDemand installation root.

You can find more information about the tablespace creation exit in the manual *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232.



iSeries Common Server migration

In this chapter we provide some suggestions and recommendations based on our experience with migration to the Common Server in customer environments.

This chapter is intended to supplement Appendix A, “Migration from Spool File Archive to Common Server” in *IBM Content Manager OnDemand iSeries Common Server - Planning and Installation Guide*, SC27-1158. Prior to reading this chapter, we recommend that you read *IBM Content Manager OnDemand for iSeries Common Server - Administration Guide*, SC27-1161, and the document *Read This First* for V5.3 or V5.4 or later releases.

In this chapter, we cover the following topics:

- ▶ Introduction to the migration tool
- ▶ Preparation
- ▶ Analysis and planning
- ▶ Migration
- ▶ Modifications to folders after migration
- ▶ Ongoing use of the Common Server
- ▶ Summary

12.1 Introduction to the migration tool

With Version 5 Release 3 of OnDemand for iSeries, a tool is available to help customers migrate from Spool File Archive to the Common Server. Before the tool was available, the only way to migrate was to re-spool all the archived reports, archive them into the Common Server, and then delete them from the Spool File Archive. With AnyStore, it was necessary to rescan or recreate the original documents and then archive and delete them from the Spool File Archive.

The migration tool makes the entire process much easier. You can use this tool to migrate users and user groups, migration policies (including optical storage groups), report definitions, and indexes. The compressed archived data itself is not moved, but the Advanced Function Presentation (AFP) resources are moved to the new integrated file system directories and the indexes and annotations are moved to new database files. The OS/400 Indexer has been enhanced to recognize migrated definitions. Also the Common Server programs have been modified, so that they can locate archived data on optical volumes, tapes, and in the /QIBM/UserData/RDARS/SpoolFile path in the integrated file system. The end result is that new data can be archived into the Common Server and users can still retrieve the migrated data.

In this chapter, we refer to document search fields as *indexes*, even though in the Spool File Archive, we often refer to them as *keys*. The terminology has changed, but it is easier for you if we are consistent in the terms we use here.

12.2 Preparation

There are two main preparation steps:

1. Set up the Common Server environment.
2. Make changes to the Spool File Archive environment.

12.2.1 Setting up the Common Server environment

First you must install the Common Server feature of OnDemand for iSeries at V5R3 or later. You must also order, load, and apply the Base, Spool File Archive, and Common Server PTFs for the OnDemand program product 5722-RD1.

Next you must create the instance QUSROND. We also recommend that you create an instance called ONDTEST for testing purposes. Refer to the *Planning and Installation Guide*, SC27-1158, for information about how to create an instance.

Edit the /QIBM/UserData/OnDemand/CONFIG/ARS.INI file, and change the port number for each instance. The default port 0 is really port 1445, which is the port used by Spool File Archive. We typically change QUSROND to use port 1450 and change ONDTEST to use port 1460.

If you want the servers to start automatically with the Start TCP/IP Server (STRTCPSVR) command, edit the /QIBM/UserData/OnDemand/'instance name'/ARS.CFG file for each instance and specify ARS_AUTOSTART_INSTANCE=1.

Start the OnDemand servers with the following command:

```
STRTCPSVR *ONDMD or CALL QRDARS/QRLMCTL *STRTCPSVRinstancename
```

The server jobs run in subsystem QSYSWRK.

To download the latest level of the OnDemand Client, go to the following Web address:

<ftp://ftp.software.ibm.com/software/ondemand/fixes/>

From this Web site, select the latest directory and the highest release level within that directory. Download the odwin32.zip file, unzip it, and run the setup.exe program.

Install the OnDemand Administrator Client on your workstation.

1. Add a server definition for each instance, specifying the port number for the instance. Log on as QONDADM, password QONDADM1.
2. The first time you logon with this ID, you must change the password.
3. Add your user ID as a system administrator, and then logoff and logon again with your ID. You must add the user profile that you will use to do the migration, or the migration commands will fail. Also, change this profile on the iSeries by using the Change User Profile (CHGUSRPRF) command to make sure it has these characteristics:
 - *ALLOBJ authority
 - Group or supplemental group profiles QONDADM, QRDARS400, and QRDARSADM
 - Locale job attributes *CCSID, *SRTSEQ, *DECFMT, *DATFMT, *DATSEP, and *TIMSEP
 - The correct locale (see Appendix D in the *IBM Content Manager OnDemand iSeries Common Server - Planning and Installation Guide*, SC27-1158)

Install iSeries Access and the latest service pack on your workstation. Then run iSeries Access Selective Setup and install the OnDemand plug-in.

Note: Updates to the plug-in are made available through PTFs to the OnDemand product code on the iSeries. After applying PTFs, you can download the changes to your workstation by removing and re-installing the OnDemand plug-in. Or, you can download the new plug-in by running **Start** → **Programs** → **IBM iSeries Access for Windows** → **Service** → **Check Service Level or Install Service Pack**.

The OnDemand Archive plug-in to iSeries Navigator is only required for OnDemand administrators; end users do *not* need this component.

Make sure that some lines are automatically added to the `[@SRV@_ONDMGINST]` stanza in the `/QIBM/UserData/OnDemand/CONFIG/ARS.INI` file. This information is provided in the Read This First Document for releases V5R3 and later.

12.2.2 Making changes to the Spool File Archive environment

We recommend that you review the migration policies and the user authorities to see if they should be changed. It is a good idea to make the current environment easier to manage and maintain before you migrate it to the Common Server.

Migration policy considerations

Typically, you need only two or three migration policies, but sometimes you might have defined several policies with small differences among them. The migration tool migrates every policy to the Common Server. Even though the Common Server works fine with many migration policies, it can be confusing to the system administrator. Therefore, it is better to have just a few migration policies that are easy to manage.

If you only use disk, you only need one migration policy in the Common Server, since the Life of Data and Indexes is specified in each application group. If you plan to use an optical library, you might still be able to use a single policy. For example, you can create a migration policy with two levels, disk pool and optical. Specify 100 days for the disk pool level and No Maximum days for the optical level. Each application group with Life of Data equal to more than 100 days is migrated to optical. Each application group with 100 or fewer Life of Data days remains in the disk pool until it is deleted by the Archive Storage Manager.

If you prefer to keep different types of documents or reports on different sets of optical volumes, then you must define more than one optical storage group and have a migration policy for each optical storage group. You might find it acceptable to keep all the archived data in the same optical storage group, or keep all the archived data on disk.

If you have several migration policies in Spool File Archive and want only a few policies, you must change each report definition to use a new policy and then use SQL to change the records in the QARLRSRT file in library QUSRRDARS. If you are not familiar with SQL or are hesitant to change production files, then it is better to migrate all the migration policies to the Common Server. If you decide to use SQL to change the migration policies referred to in the QARLRSRT file, you *must* backup both the QARLRSRT and QARLRACT files before you make any changes.

Are you using an optical library with Spool File Archive? You may decide to move your archived data back to disk or to a new IBM 3996 optical library that supports high density optical volumes (30 GB cartridges). Some business partners have programs to move Spool File Archive data from optical to disk or high density optical. It is a good idea to do this step *before* you migrate to the Common Server. Remember, the migration tool moves the indexes and AFP resources, but leaves the data in its current location.

User profile considerations

Let us review how basic security works in Spool File Archive. The QRDARS400 authorization list specifies who can access OnDemand. Typically you specify *PUBLIC *CHANGE, which means that all users can use FNDRPTRDAR to search for archived reports.

To retrieve a report, the user must be authorized to that report. By default, each report has an authorization list with the same name as the report definition. You can grant *PUBLIC *USE to the report so that all users can retrieve documents. Or you can leave the default *PUBLIC *EXCLUDE and then grant *USE authority to individual users or group profiles. Refer to Chapter 1 in *IBM Content Manager OnDemand for iSeries - Administration Guide*, SC41-5325, for information about authorizing users to Spool File Archive and to Spool File Archive reports and report groups.

Users can also be restricted by key security, which is handled in the migration tool by adding Query Restrictions for *PUBLIC and individual users to migrated application groups. For Spool File Archive reports that belong to report groups, the authority is sometimes only specified at the report group level. After all the report definitions in a report group are migrated, the migration tool migrates the report group definition to a folder. If you specify security at a report group level, you must update permissions manually in the Common Server application groups after you migrate the report definitions within the group. Or you can set report and key security at an individual report level prior to migration, and the permissions should migrate.

If you specify *PUBLIC *EXCLUDE for the QRDARS400 authorization list, the migration tool migrates only the users who are listed on the authorization lists for specific reports or who belong to group profiles that are listed on those authorization lists. If you specify anything other than *PUBLIC *EXCLUDE on the QRDARS400 authorization list, and if there are *any* report authorization lists that do not specify *PUBLIC *EXCLUDE, then *all* user profiles except those beginning with a Q are migrated.

If you do not want to migrate all the users, make sure that the authorities in Spool File Archive are adjusted to make the migration tool migrate the appropriate user profiles. The QRDARS400 and QRDARSADM profiles are always migrated.

Recommendation: At this stage, we recommend that you set your authorities in Spool File Archive to the way that you want to migrate them to the Common Server.

You may prefer to add selected users to the Common Server instead of using the tool to migrate users. If so, you can either add users manually with the OnDemand Administrator Client or write a program to add selected users. Sample programs are included in the OnDemand for iSeries Bulletin Summary from 2005 (refer to Chapter 15, “Did you know” on page 479, for more information about the bulletins).

12.3 Analysis and planning

The purpose of this phase is to analyze your current environment to determine the best method for migration to the Common Server.

You *must* use the migration tool to migrate your migration policies from the Spool File Archive to the Common Server. The tool creates two storage nodes for the migrated policy, one for the Spool File Archive and the other for the Common Server. You must have both nodes so that users can retrieve data archived in both environments. Since the data itself is not moved from its original location in the Spool File Archive, a storage node must be defined for OnDemand to find it.

Refer to the *Read This First* document. This document contains detailed information about requirements when migrating migration policies from the Spool File Archive to the Common Server. In particular, it addresses the requirements for migrating Spool File Archive migration policies that have names that match migration policies that already exist in the Common Server instance. You can find the Read This First document on the Web at the following address:

http://www.ibm.com/software/data/ondemand/pubs/readmefirst_400v5.2.pdf

You may choose to use the no-charge tool to help migrate most of your archives, but you may also re-spool and re-archive some of your reports. The analysis and planning phase of the migration is an important step that helps you to determine the best approach in your environment.

The best place to start is by submitting the Analyze Definitions (ANZDFN) command to batch:

```
CALL PGM(QRDARS/QRLRMIG) PARM('*ANZDFN' 'QUSROND' '*ALL')
```

If you have a large number of report definitions, this might be a long-running job. It is important to analyze the report definitions. After the job finishes, print the report, review it, and look for warning messages. The report tells you if you must review certain report definitions. For example, index exit programs do not migrate, so if you are using them for any of your reports, you must modify the definitions after they are migrated to the Common Server. Review the exit programs so you know how they are used. You may be able to use the same function in the Common Server without writing a post-processor program. Many customers use index exit programs to remove the dashes (-) in a social security number; you can accomplish this task easily in the Common Server by modifying the application to remove the embedded “-” character.

Be sure to review the analysis report to see if you changed the number or order of indexes in a report definition. For example, if you use Account Number for index 2 in one version of a report and index 3 in another version, a user cannot easily search for a document. In this situation, it is easier to re-spool the reports and re-archive them in the Common Server rather than to migrate a problem report.

In another situation, you might encounter a migration problem if you used DATE as the name of an index in the Spool File Archive report definition. DATE is a reserved word in the Common Server and cannot be used (in either upper or lowercase) as the name of an index. You can change the name of an index in the Spool File Archive, but not in the Common Server. If you use DATE as the name of an index, be sure to change the name before migrating that report.

It is also a good idea to review your Spool File Archive environment and get some statistics about the amount of data and types of data that must be migrated. It is helpful to know this information so that you can estimate the effort involved in the migration and track the progress.

The following list offers some suggested statistics to gather and how you can gather them:

- ▶ Number of report definitions? Display the number of records in the QARLRACT file with the following command:

```
DSPFD QUSRRDARS/QARLRACT
```

- ▶ Number of archived reports? Display the number of records in the QARLRSRT file with the following command:

```
DSPFD QUSRRDARS/QARLRSRT
```

- ▶ Number of report indexes? Display the number of records in the QARLR000PF file with the following command:

```
DSPFD QUSRRDARS/QARLR000PF
```

This file contains all the index records for reports that do not belong to a report group.

- ▶ Number of indexes for reports in report groups?

- a. From a 5250 command line, type the following command:

```
GO ONDEMAND
```

- b. Select option 1 (Report Administration) and then select option 5 (Work with Report Groups). Display each report group to get the report group abbreviation.

- c. Type the following command:

```
DSPFD QUSRRDARS/QARLRxxxPF
```

Here *xxx* is the report group abbreviation. Note the number of records in each report group index file.

- ▶ What report types are used: DOC, PAGE, NODX, UBND, or ANYS? Query the QUSRRDARS/QARLRACT file and print a listing of the report names (CDTYPE), version (VERSION), and report type (ARPTTYP).
- ▶ Which reports use key security? Query the QUSRRDARS/QARLRACT file and list the records with KEYxSECURE = 'Y' (where x = 1, 2, 3, 4, 5). When you set up coexistence (the first phase of migration), you are unable to access the Spool File Archive reports that use key security until they are migrated to the Common Server.
- ▶ What reports have been archived in the past year? Query the QUSRRDARS/QARLRACT file and list the CDTYPE, VERSION, and ARUND fields, using ARUND GE '2005-01-01' as the selection criteria. You can also use an SQL command, for example:

```
SELECT ACT_APPLICATION_ID, ACT_APPLID_VERSION,  
ACT_CURRENT_TIMESTAMP FROM QUSRRDARS/QARLRACT WHERE  
ACT_CURRENT_TIMESTAMP > '2005-01-01'
```

You must test the archival of new spooled files to these reports and versions after you migrate the report definitions. For report definitions that are not on this list, you probably only need to migrate the definitions and indexes and do not need to test archiving of the new spooled files.

- ▶ Number of indexes that are not archived on disk? Use SQL or Query to determine whether any records with IDXSTAT NE 'D' in the QUSRRDARS/QARLRSRT file. You must copy the indexes to disk (from optical or tape) before they can be migrated to the Common Server. For example, you can use the following SQL command:

```
SELECT CDTYPE, INDEX_STATUS FROM QUSRRDARS/QARLRSRT WHERE INDEX_STATUS != 'D'
```

An index recall program is included with the migration tool, but we recommend that you recall all the indexes from optical or tape back to disk *before* you start the migration. In fact, you can do this process at any time, even if you do not have the Common Server installed. Just compile and use the RTVARCIDX or RTVSPECIDX program to recall all indexes or selective indexes by optical volume. The source for these programs is found in the QSAMPLES source file in library QRDRARS. Follow the instructions to compile and run the programs, but ignore the statement in the documentation stating that no one should be using OnDemand while the program is running. It is perfectly acceptable to archive reports and have users access OnDemand while indexes are copied back to disk.

You *do* need to pay attention to the RTVARCIDX program documentation that tells you to run an SQL command to clear the INDEX_RECALL_DATE fields in the QARLRSRT file after the indexes are copied back to disk. If the fields are not cleared, then the indexes are deleted from disk as soon as the number of days specified in the Index Recall Retention field in the migration policy is reached. Then you *must* start over with the recall process.

As part of the analysis and planning phase, you must determine your migration strategy. You might have some archived reports that you want to re-spool, redefine, and re-archive in the Common Server instead of using the migration tool. Here are a few examples of reports that you might want to re-define in the Common Server instead of migrating as is:

- ▶ Reports that should have longer index lengths

For DOC reports in Spool File Archive, the maximum length for the five indexes are 25, 20, 20, 20, and 15 respectively. The maximum index length in the Common Server is 254 characters. If you have some reports where a customer name, for example, is truncated at 20 characters and you want to use 40 characters, then you might want to re-spool those report occurrences and archive them into a new definition in the Common Server.

You can get a list of the reports that currently have an index defined, that is at the maximum length, by executing the following SQL statement:

```
SELECT CDTYPE, KY1LEN, KY2LEN, KY3LEN, FL1LEN, FL2LEN FROM  
QUSRRDARS.QARLRACT WHERE KY1LEN=25 OR KY2LEN=20 OR KY3LEN=20 OR FL1LEN=20  
OR FL2LEN=15
```

- ▶ Reports that should have a different index data type

All indexes are migrated as character string fields. For example, if you prefer to define a particular report index as an integer field, you must re-spool and re-archive those report definitions using a new Common Server definition. A discussion of the data types is provided in *Appendix A of IBM Content Manager OnDemand iSeries Common Server - Planning and Installation Guide*, SC27-1158.

- ▶ Reports that are poorly defined

If you have had Spool File Archive installed for a long time, you have probably learned a few things over the years about better, more user-friendly ways to set up report definitions. The change to the Common Server might be a good time to re-define some of these reports. For example, if you installed R/DARS many years ago when it was possible to change the report type (which should never have been allowed and was later prohibited), you changed some reports from DOC to NODX, and the migration tool did not work on those reports.

- ▶ Reports that are easier to use if there are additional indexes

With the Common Server, you can define up to 32 indexes for a single application (report). You can migrate the existing report, create a new application with more indexes, and then join the two applications (migrated and new) within a single folder. But you may prefer to re-spool the existing reports and create a new definition with the additional indexes in the Common Server. That way all the archived reports will use the additional indexes, not just the new ones.

- ▶ Reports that you plan to use with the Document Audit Facility in the OnDemand Client

Using Document Audit Facility requires that you add a new field to the Application Group. Refer to Chapter 15, “Did you know” on page 479, for more information about Document Audit Facility.

- ▶ If you want to use Large Object support
- ▶ For reports that use key security

You are unable to view these reports in the coexistence (combined folder list) environment until they are migrated to the Common Server. If you have only a few of these reports, you might be able to use the migration tool to migrate them to the Common Server, and then setup coexistence and begin migrating

the other reports. Or you can gradually re-spool and re-archive the reports using key security, stop archiving each report in the Spool File Archive when it is in the Common Server, and require the users to have a separate OnDemand Client session to access these reports. When all key security reports are archived in the Common Server, you can setup the coexistence environment.

Important: You *cannot* rename migrated application groups. If you do, you will not be able to access the reports.

You cannot rename migrated application group because the archived data is not moved and is located in integrated file system or on optical by using the application group name. This is an important fact to keep in mind when considering how best to migrate.

Maybe you decide to migrate a report definition and then create a new application group with additional index fields, grouping the two application groups within the same folder for searching. You must keep the original name for the old application group and give a new name to the new application group. But that means that you must change whatever value you are using to match the application and application group name in the output queue monitor program (for example, userdata or formtype).

You may change your mind as you progress through the migration, but it is a good idea in the planning stage to think about which reports to migrate using the tool and which reports to migrate manually. It is easier to use the migration tool for all reports if possible. We suggest that you use the migration tool for:

- ▶ Report definitions that have a satisfactory number, type, and length of index fields now

These are the reports that you will be satisfied with when they are migrated to the Common Server.

- ▶ ANYS reports

If you do not use the migration tool, you must reprint and rescan documents, or write your own programs to move the indexes and retrieve and archived the data. It is much easier to use the migration tool. If you are using Kofax Ascent Capture, you must modify the document class definition to refer to the Common Server Release Script.

12.4 Migration

You can migrate with or without the migration tool. After analysis and planning, you might find that you will be able to migrate successfully using only the migration tool. However, if you have some reports that you want to redefine for some of the reasons listed earlier, you might benefit from studying the suggested approach described here.

12.4.1 Migration without the tool

This section assumes that you already created a migration policy in the QUSROND production instance or migrated one from Spool File Archive.

Report definitions

If you want to redefine report ABC in the Common Server, we present some steps that might make this process easier. The names and libraries of objects you create can be changed; this is only an example, but we found that it is easier to monitor the progress of these steps if you name the objects according to the version of the report definition you are working with. Version 01 report definitions are used by the RPTS01 query, the RPTS01 output queue, and so on.

We suggest that you use the following steps to redefine report ABC:

1. Create a query called RPTS01 in library QGPL.

The displays in Figure 12-1 through Figure 12-7 on page 413 show how you can define the QARLRSRT file, the ODATE and OSEQ result fields, and the selected fields CDTYPE, VERSION, ODATE, and OSEQ. The output to database file RPTS01 is also included.

Define the Query		
Query : RPTS01	Option : CREATE	
Library : QGPL	CCSID : 37	
Type options, press Enter. Press F21 to select all. 1=Select		
Opt Query Definition Option		
1 > Specify file selections		
1 > Define result fields		
1 > Select and sequence fields		
1 > Select records		
Select sort fields		
Select collating sequence		
Specify report column formatting		
Select report summary functions		
Define report breaks		
1 > Select output type and output form		
Specify processing options		
F3=Exit	F5=Report	
F13=Layout	F18=Files	F21>Select all
Select options, or press F3 to save or run the query.		

Figure 12-1 Defining the RPTS01 query

Specify File Selections		
Type choices, press Enter. Press F9 to specify an additional file selection.		
File	QARLRSRT	Name, F4 for list
Library	QUSRRDARS	Name, *LIBL, F4 for list
Member	*FIRST	Name, *FIRST, F4 for list
Format	SRTREC	Name, *FIRST, F4 for list

Figure 12-2 Specify File Selections display

Define Result Fields

<u>Field</u>	<u>Expression</u>
ODATE	substr(ONAME,1,8)
OSEQ	substr(ONAME,10,3)

Figure 12-3 Define Result Fields display

Select and Sequence Fields		
Seq	Field	Text
10	CDTYPE	Report Type
20	VERSION	Version
30	ODATE	substr(ONAME,1,8)
40	OSEQ	substr(ONAME,10,3)

Figure 12-4 Select and Sequence Fields display

Select Records				
AND/OR	Field	Test	Value	(Field, Number, 'Characters'..)
	CDTYPE	EQ	'ABC'	
AND	VERSION	EQ	'01'	

Figure 12-5 Select Records display

Figure 12-6 Output type

```

Define Database File Output

Type choices, press Enter.
(The printed definition shows the output file record layout.)

File . . . . . RPTS01           Name, F4 for list
    Library . . . . . QGPL          Name, F4 for list
    Member . . . . . *FILE         Name, *FIRST, *FILE, *ALL,
                                    F4 for list
Data in file . . . . . 2          1=New file, 2=Replace file

```

Figure 12-7 Database output

2. Run the query, which creates a database file called RPTS01. There is one record in the file for each report occurrence of Version 01 for the ABC report definition. See Figure 12-8.

```

Define Database File Output
Exit this Query

Type choices, press Enter.

Save definition . . . Y           Y=Yes, N=No
Run option . . . . . 2            1=Run interactively
                                2=Run in batch
                                3=Do not run

For a saved definition:
Query . . . . . RPTS01          Name
    Library . . . . . QGPL        Name, F4 for list

```

Figure 12-8 Running query RPTS01

3. Create an output queue called RPTS01 in library QGPL.
4. Start and end a monitor on this output queue with the following command:
`STRMONOND OUTQ(RPTS01) ENDMON(*NOINPUT)`
This command creates and attaches a data queue to the output queue.
5. End the monitor with the following command:
`ENDMONOND TYPE(*OUTQ) OUTQ(RPTS01)`

6. Write and compile a CL program as shown in Figure 12-9.

```
***** Beginning of data *****
PGM
DCLF      FILE(QGPL/RPTS01)
LOOP:
RCVF
MONMSG    MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
PRTRPRTDAR REPORT(&CDTYPE) VERSION(&VERSION) +
             RPTDATE(&ODATE) RPTSEQ(&OSEQ) +
             PRINTER(*OUTQ) OUTQ(QGPL/RPTS01) SBMJOB(*NO)
GOTO      CMDLBL(LOOP)
END:      ENDPGM
***** End of data *****
```

Figure 12-9 PRTRPRTS01 CL program

7. Submit the CL program PRTRPRTS01 to batch, using job description QRDARS/QRDARS400. The job re-spools all ABC Version 01 report occurrences.
SBMJOB CMD(CALL PGM(QGPL/PRTRPRTS01)) JOBD(QRDARS/QRDARS400)
8. Use the Report Wizard in the OnDemand Administrator Client production instance (QUSROND) to create an application group, application, and folder for this report. See Figure 12-10 on page 415.
Be sure to allow the capability to add multiple applications to the same application group, which is the way you can have different versions of a report as the report layout changes over time. Remember that it is easier to automate the archiving of spooled files if you use the same name for the application group and the application. The name should match the value you look for in the STRMONOND command (for example, userdata or jobname). If you use the same name for both application and application group, you do not have to match each name with a spooled file attribute.

Note: You cannot use the Report Wizard if you want to use the Document Audit Facility, but it is easy to create the application group, application, and folder separately.



Figure 12-10 Application version

The definition that you just created is now associated with application group ID 01. See Figure 12-11.

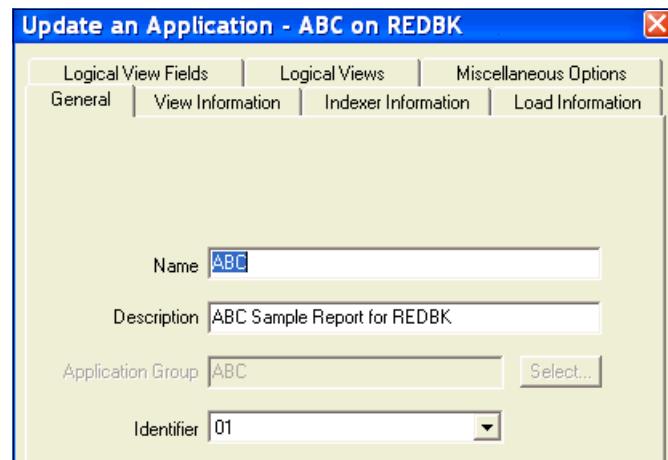


Figure 12-11 Update an Application display

9. Update the ABC application group. See Figure 12-12.

On the Storage Management panel, change the Life of Data and Indexes to be the number of days that you want the archived data and indexes to exist. In the Permissions panel, give Logical Views permission to *PUBLIC.

We found that in most cases that this is an easy and effective way to handle permissions, giving access to *PUBLIC at the application group level and then controlling permissions at the folder level. When a user logs on to the OnDemand Client, they only see a list of folders that they are authorized to access, so it does not matter if they are authorized to application groups if they cannot see them.

Important: Do not use this technique if you plan to have a single folder that contains multiple application groups and plan to allow different permission levels for different users to application groups within the folder.

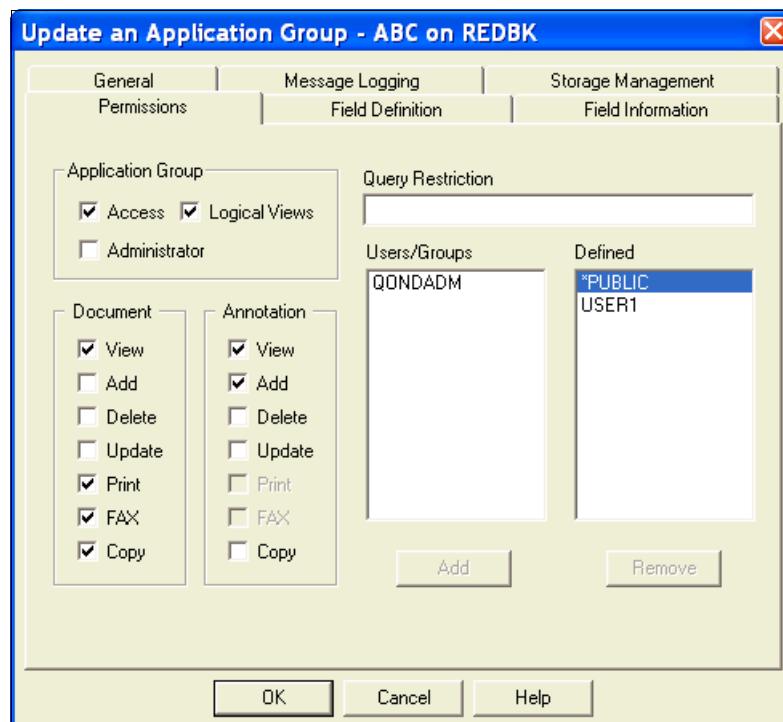


Figure 12-12 Update an Application Group

10. Use the Add Report to OnDemand (ADDRPTOND) command to test the archival and retrieval of one of the spooled files.

11. When you are satisfied with the results of your new report definition, delete the spooled file used as a test (since it is already archived). Then archive the other ABC Version 01 reports into the Common Server by starting a monitor on the RPTS01 output queue (STRMONOND):

```
STRMONOND OUTQ(RPTS01) APPGRPSRC(*USERDATA)
```

This method of re-spooling and re-archiving reports only works if the posting date was extracted from the report in Spool File Archive. If the posting date in the report definition is blank, then the date used is the date that the report is archived, so the results in the Common Server will not be the same.

Sometimes you may use a blank posting date if the date within the report is the same date as it was archived. In this case, you can define an index for the date by selecting the date field in the graphical indexer in the Common Server.

12. If any spooled files go to the QUSRARDARS/ONDERR output queue, look in the System Log folder in the OnDemand Client to determine why the report failed to load and fix the problem.
13. When all the Version 01 reports are archived, check to see if there is a Version 02 of the ABC report definition in Spool File Archive. If there is, follow these steps:
- Update the ABC application and rename it ABC-01.
 - Update the ABC application group and add a value for the Version application ID field. See Figure 12-13.

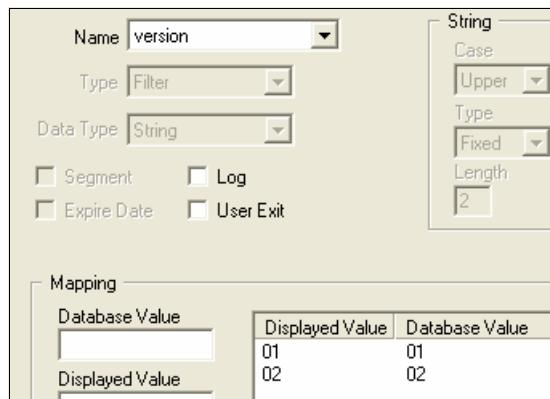


Figure 12-13 Adding an application ID value

- Copy the ABC-01 application into a new application called ABC, with identifier 02.

- d. Copy the RPTS01 query to a new query called RPTS02, changing it to select ABC reports where VERSION EQ '02' and creating a new database file called RPTS02.
 - e. Copy the RPTS01 CL program to a RPTS02 CL program, which reads and processes records from the RPTS02 file.
 - f. Create a RPTS02 output queue. Then start and end the monitor to create and attach a data queue.
 - g. Run the query and CL program to re-spool the Version 02 ABC reports.
 - h. Use a sample Version 02 spooled file to modify the indexer parameters for the ABC application. Test the archival and retrieval of the spooled file.
 - i. When you are satisfied with the new definition, delete the spooled file that you used as a test and start the monitor on the RPTS02 output queue.
14. If there are other versions of report ABC, define and archive them as well, using the method described in the previous step.
15. After archive all report occurrences for each version of report ABC into the Common Server, delete the stored reports from Spool File Archive Create and submit a CL program called DLTRPTS01. See Figure 12-14.

```
***** Beginning of data *****
      PGM
      DCLF      FILE(QGPL/RPTS01)
LOOP:   RCVF
        MONMSG  MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
        DLTRPTRDAR REPORT(&CDTYPE) VERSION(&VERSION) +
                    RPTDATE(&ODATE) RPTSEQ(&OSEQ) SBMJOB(*NO)
        GOTO     CMDLBL(LOOP)
END:    ENDPGM
***** End of data *****
```

Figure 12-14 DLTRPTS01 CL program

Note: Be *sure* that you successfully archive all the files in the Common Server before you delete them from Spool File Archive.

16. Clear the QUSRRDARS/ONDPROC output queue.

This process can be modified so that you can select several different reports with the same version number and send all of the spooled files to the RPTS01 output queue. In doing so, be careful to separate the different versions of the reports because you probably created different versions for a reason. The spooled files are unable to use the same definition.

Note: If you change the report definitions *without* creating new versions, then the re-spooling and re-archiving process will be more difficult.

We suggest that you re-spool all of the occurrences for a report, and then set up a definition using the oldest spooled file first. Start the output queue monitor and see which spooled files go to the error queue; use those files to set up new applications (with new application ID fields) within the application group.

Also review the results of successful loads. The location of a field might have changed and the report loads fine, but the beginning or ending location of the index field is incorrect. In this case, the report can store successfully, but the index values are wrong. To verify that the values that are stored are correct, search the OnDemand client for the report to see that the values in the hit list look correct.

For reports where you have continued to modify Version 01, it is easier to use the migration tool.

Users

You may prefer to add selected users to the Common Server instead of using the tool to migrate users. If so, you can either add users manually with the OnDemand Administrator Client, or write a program to add the selected users. Sample programs are included in the OnDemand for iSeries Bulletin Summary from 2005 (refer to Chapter 15, “Did you know” on page 479, for more information about the Bulletins).

If you do not use the migration tool to migrate users, you must still create an ADMIN user ID for the local server instance used by the migration tool. This ID is created automatically by the *MGRUSR (migrate users) option of the tool, so you can create this ID in either of these two ways:

- ▶ Run the *MGRUSR option to the ONDTEST instance. This creates the ADMIN user ID in the local server used by the migration tool.
- ▶ Create the USER.TBL file in the /OND_MIG_INST/TABLE directory in the iSeries integrated file system and add these lines to the file:

```
EDTF STMF(' /OND_MIG_INST/TABLE/USER.TBL ')
[ADMIN]
UID=79999
PASSWD="ssjbENv1dbaoA"
ADMIN=4
PID=0
LAST_UPDATE=-1
TIMEOUT=0
```

12.4.2 Migration with the tool

The analysis and planning phase should help you decide the best approach in your particular environment. In this section, we present a suggested approach for migration using the tool. You can find the commands and exact instructions in Appendix A of *IBM Content Manager OnDemand iSeries Common Server - Planning and Installation Guide*, SC27-1158.

1. Migrate users to the ONDTEST instance.
2. Use the export feature in the Administrator Client to export users from the ONDTEST instance to the QUSROND production instance. You can select all users or only a few users. You can do this step at anytime during or after the migration.
3. Migrate the migration policies to the ONDTEST instance and to the QUSROND instance.
4. Migrate all the reports using key security, either by using the migration tool or by re-spooling and re-archiving the reports (refer to page 406 for considerations on key security).
5. Set up coexistence for the QUSROND instance.
6. Create an ONDTEST output queue (we typically create it in the QGPL or QUSRSYS library) and start a monitor with the command:
`STRMONOND TYPE(*OUTQ) OUTQ(ONDTEST)`
7. Create an output queue that can be used for archiving spooled files in the QUSROND production instance. This might be an output queue that is only used temporarily. When you finish migrating, you may prefer to use the output queues for the Common Server that you are currently using in Spool File Archive. That way you do not have to change your business application programs or printer files to direct the spooled files to new output queues.
8. Review the results from the query in the Analysis and Planning phase that listed the reports that were not archived during the past year. You should be able to migrate these reports and indexes and test retrieval of the data, without needing to archive new spooled files. This presumes that if report definitions have not been used within the last year, they are no longer used for archiving spooled files. If you have some year-end reports that are in this list, test archival of the new reports when they are available.
9. Create a data area to suppress all the Qshell job logs that are created when doing index migration:
`CRTDTAARA DTAARA(QUSRRDARS/QRLRMIGJOB) TYPE(*CHAR) LEN(1)`
10. Migrate and convert the definitions and indexes for these no-longer-used reports. Since you will not archive new spooled files, you can migrate directly to the QUSROND instance. Test retrieval of a sample of each report.

11. After the indexes are migrated, when you view the folder list in the OnDemand Client, these reports should no longer include the co-existence "suffix" (the server name in the QUSROND configuration file).
12. For each currently used report definition you plan to migrate, use the PRTRPTRDAR command to re-spool a sample report using the highest version of the report. Keep these spooled files in a separate output queue that you create to hold the files prior to migration and testing (for example, CRTOUTQ ONDMIGTEST).

```
PRTRPTRDAR REPORT(APCHECKS) VERSION(03) RPTDATE(20060209) RPTSEQ(001)  
PRINTER(*OUTQ) OUTQ(*LIBL/ONDMIGTEST)
```

13. Depending on the number of reports, you may choose to migrate definitions individually or generically (all reports beginning with A*, for example), or by report group. Regardless of what you decide, you must follow these steps:
 - a. Migrate the report definition to ONDTEST; an application group, application, and folder are created.
 - b. Modify the application if necessary, for example, if an index exit program was used for this definition in Spool File Archive.
 - c. Move the sample spooled file for this report from the ONDMIGTEST outq to the ONDTEST outq. Be sure the spooled file is in RDY status and the outq monitor is active. See where the spooled file goes, to the ONDPROC or the ONDERR outq (in library QUSRNDARS).
 - d. If the report is loaded successfully, test retrieval of it in the OnDemand Client. Compare the results (such as number of documents and the actual indexes) with the original report in Spool File Archive.

The Common Server might deliver different results because Spool File Archive does not consider a blank a change in value and the Common Server does. For example, if a report definition in Spool File Archive specifies a change in department number as the segmentation criteria, a page with spaces in the department number field is part of the previous segment. In the Common Server, spaces are considered a valid index value, so a separate document is created when the department number changes, even when the new value is spaces. In this example, you have a different number of documents for Spool File Archive and Common Server.

- e. Make the necessary changes to the application.
- f. When you are satisfied that you can archive and retrieve data with the migrated application, export the application group and folder to the QUSROND instance from the ONDTEST instance. You can do this easily using the Administrator Client.

- g. Convert the report definition. From this point on, this report definition can no longer be used for archiving reports in Spool File Archive.
- h. Change your business application programs to start directing the spooled files used by this report definition to a production output queue monitored by the Common Server.

Or you can change the Start Monitor for OnDemand (STRMONRDAR) command you are using in Spool File Archive to direct reports that are in error to the Common Server output queue. (Spool File Archive does not find reports that are converted so they go to the error output queue.) That means you do not have to change your own applications. When you successfully migrate all reports, you can stop using the STRMONRDAR command and start using the STRMONOND command on the same output queues (after first detaching and then deleting the associated data queues).

- i. Migrate the report indexes. *Be sure* to specify QUSROND as the instance name. You do *not* want to migrate the indexes to a test instance; otherwise the users will not be able to retrieve the reports. You can migrate the indexes for a report by a specific date range, or you can use the parameters *AVAIL and *CURRENT to migrate all the indexes for the report.
- j. Remove the report indexes from Spool File Archive.

14. After all reports and indexes are migrated, reorganize the index files used by Spool File Archive. By doing this step, you free the space used by deleted records.

Note: You must *always* modify the application indexer parameters in a migrated PAGE report, which is referred to as a *transaction report* in the Common Server. This type of report has a beginning and ending range of index values for every 100 pages. (With the Common Server, there is not a 100-page limit for document size.) A mask is used to find these values in the Common Server, but you must specify the type of characters to search for in the mask. For more information about how to define transaction reports, see *IBM Content Manager OnDemand for iSeries Common Server - Indexing Reference*, SC27-1160.

It is easy to track the progress of the migration by using Query or SQL. There is a record for each report definition in the QARLRACT file in the QUSRRDARS library. The value for EXTRAFIELD1 in this file is changed as you successfully complete the migration steps for a report. There is a record for every report occurrence in the QARLRSRT file in library QUSRRDARS. The value for RESTIND in this file is changed to M whenever all the indexes for that report occurrence have been migrated to the Common Server.

You can find more details about tracking the status and progress of the migration in the News and Tips from the 2005 e-mail bulletins. To access these bulletins, search on “Bulletins” on the IBM OnDemand for iSeries Support Web site at the following address:

<http://www.ibm.com/software/data/ondemand/400/support.html>

12.5 Modifications to folders after migration

There are several helpful features in the Common Server to assist users after migration. To take advantage of these feature, you might want to make the following changes after migration:

- ▶ Remove the GROUPMAXPAGES=100 line from the Indexer Parameters in the applications. Spool File Archive has a limit of 100 pages per document, but there is no limit in the Common Server.
- ▶ Change the Sequence Number and Application Identifier fields to a value of 0 for Query and Hit List in the folders. The users do not normally want or need to see these values in the document list (hit list). Figure 12-15 shows the folder setting before making the changes.

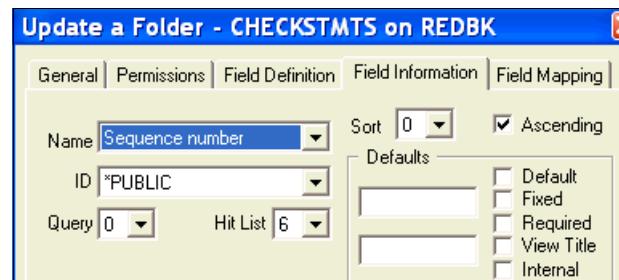


Figure 12-15 Folder settings before making changes

Figure 12-16 shows the folder setting after the changes.

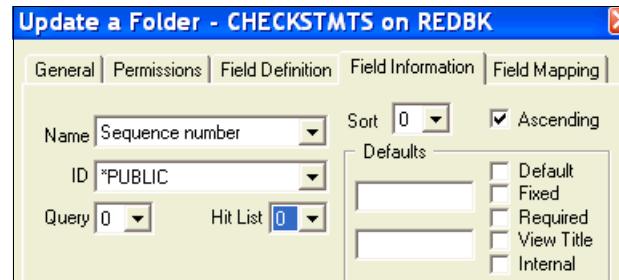


Figure 12-16 Folder settings after making changes

- ▶ Change the date formats and search interval to defaults that are acceptable for the users. Be consistent and modify all the folders to use the same defaults, if possible. The migrated definitions use a %m/%d/%y (four-digit year) format and the users might prefer a two-digit year for searching. See Figure 12-17.

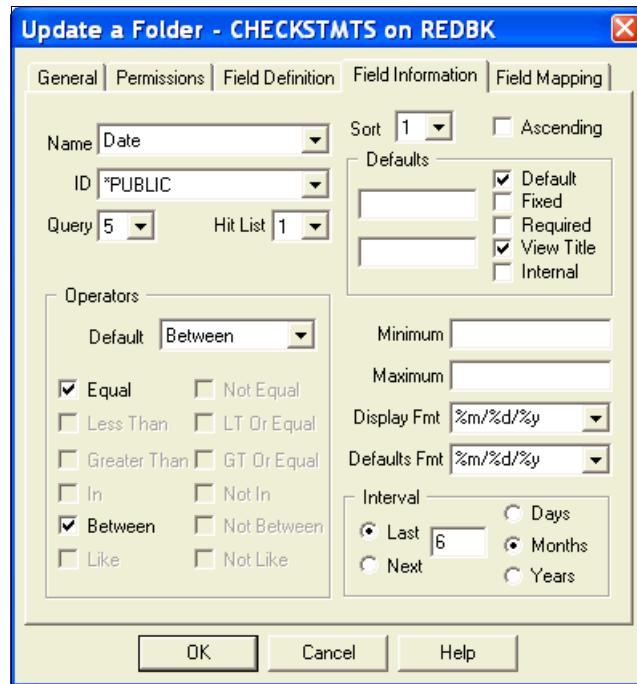


Figure 12-17 Default settings for updating the folder date

- Group multiple application groups into a single folder if the indexes are the same. That way the users have fewer folders to search.

For reports that were defined as NODX (no index) in Spool File Archive, we found that it is helpful to group similar-type reports into a single folder. For example, group all the financial reports into a new folder called FINRPTS. Only two indexes are needed, Report Date and Report Name. See Figure 12-18.



Figure 12-18 Adding a folder

Map the Report Date to F_01 (posting date from Spool File Archive) and Report Name to F_00 (application identifier or version from Spool File Archive). See Figure 12-19.

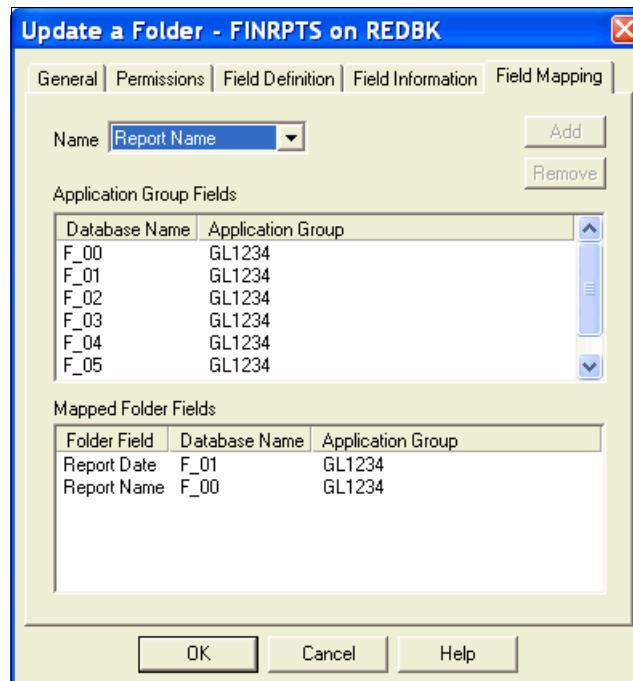


Figure 12-19 Updating the FINRPTS folder

Update the Application ID field in the application group so that the descriptive report name is displayed instead of the internal database value. See Figure 12-20.

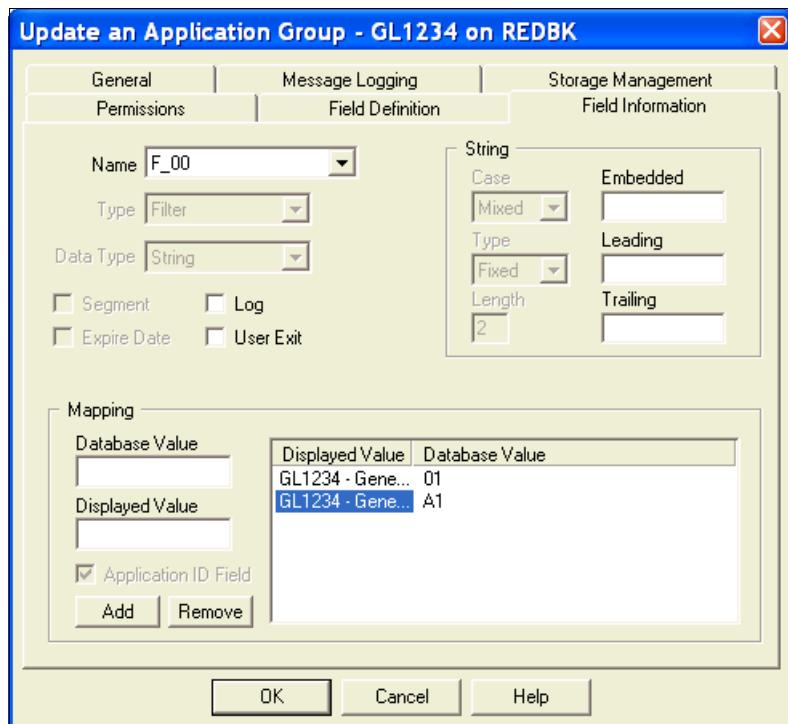


Figure 12-20 Update application group

The displayed value can be the same for all Application ID Field values for the application group. Now when users search for documents in the folder, they can select a single report to display. See Figure 12-21.

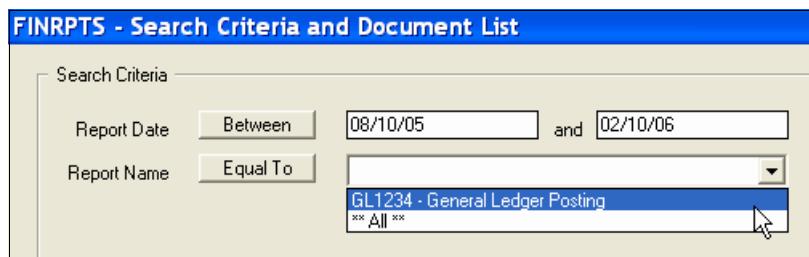


Figure 12-21 Folder search

12.6 Ongoing use of the Common Server

Remember that you must continue to run the Spool File Archive Report Management Cycle (RMC) so that the migrated data expires. Reports that have been migrated continue to use Spool File Archive migration policies to manage their life cycles.

Be sure to automate the OnDemand jobs. We recommend that you add the following tasks to the iSeries Job Scheduler:

- ▶ STRTCPVR *ONDMD
- ▶ STRMONOND
- ▶ STRDSMOND

Also, before you back up the OnDemand integrated file system directories, you must unmount the file system if you are using a disk pool. If you are using ASMASP01 in instance QUSROND, you use the following command:

```
UNMOUNT TYPE(*UDFS) MFS('/dev/QASP01/ONDEMAND_QUSROND_PRIMARY_01.UDFS')
```

Two tasks cannot be automated. First you must review the error output queue each day to see if any reports failed to archive. This step is also necessary in Spool File Archive. Second review the QPRLCASM1 report, which is the status report that is created whenever Archive Storage Manager is run. The default location for this Archive Storage Manager report is output queue QRDARS400 in library QRDARS, but the QPRLCASM1 printer file can be modified so that another output queue is used instead.

12.7 Summary

Both the Spool File Archive and Common Server are designed to help customers archive and retrieve spooled files and other documents. However, the directory and database structure of the two product features are so different that it is remarkable that it is even possible to migrate from one environment to the other.

The migration tool is a complex set of programs that works well in making this migration possible. However the migration process cannot be fully automated; an OnDemand administrator must review each step of the process to ensure that you achieve accurate results.

Since knowledge of both the Spool File Archive and Common Server is required to understand the migration process, we recommend that you acquire formal education about the Common Server from an experienced IBM Business Partner. You may also choose to have an IBM Business Partner handle the migration for you. When the environment is migrated, there is no longer a need to understand or work with the migration tool.

The Common Server offers a lot of advantages for both users and administrators. Learn as much as you can about this product so you can make enhancements to both migrated and new applications and take full advantage of the new features in the Common Server.



Solution design and best practices

In this chapter, we re-iterate some of the concepts that you have learned in the previous chapters of this IBM Redbooks publication and help you learn how to design an OnDemand solution for performance and ease of use.

We include the following topics:

- ▶ Designing a winning solution
- ▶ Best practices

Note: The contents of 13.1, “Designing a winning solution” on page 430, are contributed by the OnDemand development and support lab personnel. The various best practices tips were collected from both the OnDemand lab personnel and OnDemand practitioners in the field.

13.1 Designing a winning solution

Your company has decided to purchase the IBM Content Manager OnDemand (OnDemand) solution and has put you in charge of the project. You have gone to OnDemand University; you have attended the workshops; you have spent a great deal of time out on the OnDemand Web Support site; and you have read first part of this OnDemand IBM Redbooks publication.

When your supervisor asks you if you are prepared to design an OnDemand solution that performs well and is easy to use, how will you answer?

We intend to help you answer that question positively and confidently in this section. Although we do not discuss every possibility available for correctly designing an OnDemand solution, we provide you with an idea of what you should think about while creating your solution.

Most people who are new to OnDemand understand that OnDemand stores documents and provides the ability to retrieve these documents on a PC or a Web browser. They might also understand the basic functions of the individual components or have the ability to create an application that stores a document type.

However, most people who are new to OnDemand, and even some who have been using OnDemand for a while, have no idea if the way they use their OnDemand solution is easy to use or performing well. The OnDemand users have grown accustomed to the way things are and nobody has suggested ways to make improvements.

To help you learn how to design an OnDemand solution, you first must understand what OnDemand is. This overlaps somewhat with the information that we presented in the earlier chapters; however, it is an important concept for us to go over here again in a slightly different approach.

13.1.1 What is OnDemand?

The power of OnDemand is to provide users with specific information that they are looking for in the shortest amount of retrieval time. This can be accomplished with a little forethought and understanding of how users use the data.

In simple terms, *OnDemand* is an electronic version of your local library back when your local library required you to use a manual card catalog system. Table 13-1 shows how your local library and *OnDemand* compare.

Table 13-1 Local library compared with *OnDemand*

Local library	<i>OnDemand</i>
Page in a book	Page in a document
Chapter in a book	Document inside the store object
Book	Stored object
Card in a drawer	Row in database table
Card catalog drawer	Database drawer
Card catalog drawer	Folder

At the library, when you want to research a specific chapter in a book, you go to the card catalog cabinet, open a specific card catalog drawer, find a specific card that tells you where the book is located, find the book, and then turn to the chapter you want to see.

With *OnDemand*, you select a folder that is attached to certain database tables. You then enter your search criteria. When you press Enter, you see a document *hit list*, which is a copy of specific rows in a database table. When you select one of these rows, you retrieve the document that is of interest to you.

This is interesting but how is it related to performance and ease of use of *OnDemand*?

Let us go back to your local library. You know which book you want to see, but when you reach the card catalog cabinet, you see that four card catalog drawers have the exact same label on the front. To determine where your book is located, you must search all four of the drawers. This is obviously much slower than searching a single card drawer.

The same is true for *OnDemand*. If your search criteria must go across multiple application group tables, your query takes longer to complete. This means that your system works harder to perform the query, and your user is waiting longer for results.

Your mission as an OnDemand solution designer is to determine the best possible way to design your application so that OnDemand is not searching across multiple application group tables. Multiple application group tables can come from the same application group, or they can come from multiple application groups.

Before you start designing your application, you must understand the four pieces of an OnDemand application that are going to mean the most to your design:

- ▶ Application

The OnDemand application is for *indexing*. It gathers, from the data, the information that OnDemand needs to insert database rows and the resources that OnDemand needs for the data being loaded. Resources can be anything from logos, special fonts, page definitions, and form definitions. The application also contains information for viewing and printing the data.

- ▶ Application group

The application group is a *table builder*. It tells you which columns you must add to a table when the table is built. While loading, it ensures that the application provides the correct field (column) information to successfully insert a row.

- ▶ Folder

The OnDemand folder provides users with a *view of the database*. It provides a user with a fill-in-the-blank approach to SQL and determines which database tables that OnDemand queries.

- ▶ Storage

Storage refers to the cache storage and optical or tape storage using Tivoli Storage Management. This is where your data resides. When you retrieve a document, you obtain it from either cache or from Tivoli Storage Manager.

Although many more variables are involved in the overall OnDemand solution, a successful OnDemand solution design is most concerned with these four variables.

13.1.2 Three-step approach in solution design

If you use OnDemand for a single report type, designing for a successful OnDemand solution is easy. For most customers, a normal OnDemand solution is used daily by thousands of people who retrieve hundreds of data types.

Therefore, designing for performance and friendliness takes a three-step approach:

- ▶ Step 1: Determine how data will be retrieved.
- ▶ Step 2: Make data retrieval efficient.
- ▶ Step 3: Design folders for performance and usability.

Step 1: Determine how data will be retrieved

OnDemand finds documents based on the information that you collect in your database. It is important to understand how users search for data. When a user enters a query, you want to be able to return meaningful information to the user as quick as possible. This means that you only search across a limited number of database tables and you only return a limited number of query matches. At best, there is only a single query match.

When a user opens a folder, OnDemand already starts a database query. Selecting a folder limits the number of application group tables that OnDemand searches against. We discuss this part of the query in “Step 3: Design folders for performance and usability” on page 435.

A folder provides an interface that allows a user to query database tables. The fields of a folder correspond to database index and filter fields. If a segment field is available, this narrows the tables that OnDemand searches. A folder can also have a server-based text search field. Using a text search field is the worst way to search for data. Whenever possible, avoid providing text search fields for the OnDemand users.

When a user requests a query, the following actions occur from a high-level perspective:

1. If part of the query is a segment field, OnDemand selects only the tables that meet this criteria.
2. If index fields are part of the query, OnDemand searches the indexes of the tables selected by the segment search to choose the matching table rows.
3. If filter fields are chosen, OnDemand looks at the selected rows to narrow the resulting rows to only those rows with matching filter fields.
4. OnDemand returns the matching query rows, in a hit list, to the user.

Although it is tempting to make all fields the index fields, you must understand the trade-off. When you create an index, you take information already stored in your database table and use the database space to add that information again in a separate table, as well using the space for the overhead involved in creating an index. If all of your table fields are indexes, your database is unnecessarily large without additional benefit.

Having too many fields, index or filter, is also not a good idea. Each database column that you tell the application group to create for a table is additional space needed in your database. You must only create fields (database columns) for the items that you need to search against. At the minimum, provide at least one index field. This field should define the most unique value in the data, such as customer number, Social Security Number, and phone number. It should also be a field that a user normally searches on.

This is why it is imperative to understand how the users plan to search for documents. If your users generally search on a single field, you obviously only need that field. If 80% of your users query using three fields and another 20% need other fields as well as the three fields used by others, it might be best to have only three indexes and leave the other fields as simply filter fields.

An *application group table* is limited by the number of rows that it can have. You set this limit within the application group properties. By default, an application group has 10 million rows. When you have stored the 10 million rows, OnDemand closes the first table and opens a second table. By doing this, you degrade performance. At the same time, if you set your row limit too high, performance also degrades because it takes too long to look through the table to match your query.

This is where the *segment fields* come in. You should always specify a segment value to improve performance, usually a report date or statement date. If one does not exist in the report, you can always use a load date by specifying it in the application. This value should be chronological to provide the best segmentation. A segment field allows you to limit the number of tables you choose to search. If your segment is “load date” and you fill a table four times per year, you can limit the search to a single table simply by adding the month and year to the search. At most, a month can carry over into a second table. However, this successfully narrows the search simply by narrowing the tables you search across.

Step 2: Make data retrieval efficient

To make data retrieval efficient, you must understand how often users retrieve data. Data that is retrieved frequently should remain in the cache storage until it is no longer needed by 90% of the users. Data that is retrieved infrequently can be stored on the long-term storage, such as Tivoli Storage Manager.

Cache storage is the fastest means to deliver data to your users. When the demand for the data is no longer high, the data should be moved from the cache to long-term storage; users can still retrieve the data from long-term storage. Having a disk or tape placed in a drive and then having the drive spin up and deliver the data to your user takes considerably longer time than retrieving it from cache. It gets worse if too many people retrieve the data from Tivoli Storage Manager and a drive is not currently available to fulfill the retrieval requests.

There are some limitations regarding the amount of the cache storage you can use. The general rule of thumb is to keep the data in cache for as long as possible. Since you paid for those hard drives, you should use them. In general, you do not want your users to wait for the data to be delivered from long-term storage.

Step 3: Design folders for performance and usability

A folder can have a single application group or multiple application groups assigned to it. The best situation is when there is only a single application group assigned to a folder. This, however, is not always a practical situation.

We provide some basic OnDemand principles to help you design applications, application groups, and folders for performance and usability:

- ▶ Data objects should use the *same application* when the data is of the same type and the field information is located in the same place for each data object. You do not need to create a field to identify the report or data type, and the resources are the same.
- ▶ Data objects should use the *same application group* but different applications if the field information is the same but is not located in the same place for all data objects. The resources are entirely different and the data types are different.

You can create an application ID field to determine which application the data comes from. This means that it is possible for AFP, JPG, LINE, PDF, and TIFF data to reside within the same application group as long as the indexed information is the same.

- ▶ Before you design an application group, always consider if there is a chance when you have *more than one application load to the application group*. With the application ID, you have the ability to expand the field to include more applications. If you do not add an Application ID field, you cannot go back and add this field later.
- ▶ Application groups can reside in the *same folder* as long as there are common search fields within each of the application groups. In some cases, users do not have equal access to all the application groups. In this case, you might be concerned with the query retrieval time or expect that each application group has a large quantity of tables that a segment search will not narrow for you. In this case, consider having an *individual folder for each application group*. You can also limit the number of application groups searched in a folder by using user and group permissions.

- ▶ A folder should be your first query as well as your *first query restriction*. Users should only see folders that contain application groups with information they use. A folder should only contain application groups or portions of application groups that are similar and can logically be grouped together; users should only see folders that pertain to their jobs. For example, a payroll folder should not include inventory documents. You can limit the number of folders that users see in the folder list by using user and group permissions.

In summary, when designing your solution, you want to accomplish the following plan:

- ▶ For applications: one to many data objects
- ▶ For application groups: one to many applications
- ▶ For folders: one to one, or one to few, application groups

13.1.3 Solution design case study

Now that we have the basics, we go through a case study and see the type of OnDemand solution that you design.

Your company has the following six reports that they want to store in OnDemand:

- ▶ Balance Sheet - AFP Data
- ▶ Sales Detail Report - Line Data
- ▶ Inventory Detail Report - AFP Data
- ▶ Transaction Detail Report - Line Data
- ▶ Income Statement - PDF
- ▶ Payroll Ledger - AFP Data

Example of a bad solution design

Using only this information, a simple, but poor design looks like this:

- ▶ Create an application for each report.
- ▶ Create an application group for each application.
- ▶ Make all fields index fields.
- ▶ Create a single folder accessed by all users that contains all six application groups.

Any time a user searches this single folder using a common index field, OnDemand searches across all six application groups. Since there is no segment field, OnDemand searches across all of the tables in all six of the application groups. To make matters worse, because every field is an index field, the database is quite a bit larger than it needs to be.

Example of a good solution design

With a little more research, you can determine how users use the data:

- ▶ Balance Sheet - AFP Data
 - Users usually query on account numbers and dates.
 - Users occasionally query on account descriptions.
 - Executives and accountants use this report.
- ▶ Sales Detail Report - Line Data
 - Users usually query on account numbers and dates.
 - Users occasionally query on account descriptions.
 - Executives, accountants, and salespeople use this report.
- ▶ Inventory Detail Report - AFP Data
 - Users usually query on product numbers and dates.
 - Users occasionally query on product descriptions and transaction types.
 - Executives, accountants, inventory control personnel, and salespeople use this report.
- ▶ Transaction Detail Report - Line Data
 - Users usually query on account numbers and dates.
 - Users occasionally query on account descriptions and transaction types.
 - Accountants use this report.
- ▶ Income Statement - PDF
 - Users usually query on account numbers and dates.
 - Users occasionally query account descriptions.
 - Executives and accountants use this report.
- ▶ Payroll Ledger - AFP Data
 - Users usually query on employee numbers, last names, dates, and social security numbers.
 - Users occasionally query on first names and departments.
 - Accountants, human resources personnel use this report.

Using this information, we can design a better OnDemand solution.

The Payroll Ledger is the only report that is used by the human resources personnel; therefore, we design the following items into the solution:

- ▶ Folder “Payroll Ledger”
- ▶ Application group “payledge”
 - Segment Field: Date
 - Index Field: employeenum, lastname, ssn
 - Filter Field: firstname, dept

- ▶ Application “paypledge”

The Inventory Detail Report is the only report viewed by the inventory control personnel; therefore, we design the following items into the solution:

- ▶ Folder “Inventory Detail Report”
- ▶ Application Group “invreport”
 - Segment Field: date
 - Index Field: prodnum
 - Filter Field: proddescr, transtype
- ▶ Application “invreport”

The Transaction Detail Report is similar to the remaining three reports, except it has a requirement of the transaction type information being occasionally query on. This field is not needed by the other accounting reports. Because of this, we design the following items into the solution:

- ▶ Folder “Transaction Detail Report”
- ▶ Application Group “transdtl”
 - Segment Field: date
 - Index Field: acctnum
 - Filter Field: desription, transtype
- ▶ Application “transdtl”

Finally, we have three reports, Balance Sheet, Sales Detail Report, and Income Statement left. These reports have different data types, but they have the same query needs. The only thing we have to watch for is that the Sales Detail Report is the only one that is used by the salespeople. However, these reports are good candidates for a single application group. We design the following items in the solution:

- ▶ Folder “Executive Reports” “Sales Detail Report”
 - This will be restricted by the application ID.
- ▶ Application Group: “execreport”
 - Segment Field: date
 - Index Field: acctnum
 - Filter Field: acctdescr, application ID
- ▶ Application “balsheet” “salesrpt” “incomestmnt”

This solution requires six applications, four application groups, and five folders with access controlled by five groups. Each query searches across a minimum of a single table. Most user searches will be index scans (via index fields), as opposed to table scans (via filter fields). Because we have the date field listed as our segment date, if we load the data in the date order and require users to enter a date, we have an excellent opportunity to restrict the queries to the fewest tables possible.

This solution is not the only possibility for an excellent OnDemand solution design. There are several things we are able to do, such as query restrictions, user group restrictions, and application group permissions. The OnDemand support Web site provides an excellent resource to assist you with other design possibilities:

- ▶ DB2 Content Manager OnDemand for Multiplatforms product support
<http://www.ibm.com/software/data/ondemand/mp/support.html>
- ▶ DB2 Content Manager OnDemand for iSeries product support
<http://www.ibm.com/software/data/ondemand/400/support.html>
- ▶ DB2 Content Manager OnDemand for z/OS product support
<http://www.ibm.com/software/data/ondemand/390/support.html>

As the designer of the OnDemand Solution, you will likely be presented with a wide variety of reports to archive. They will not all be line data or Advanced Function Presentation (AFP) data, and they will all have different query needs. The best solution design that you can achieve requires understanding of users who use these reports and how the reports will be queried. Detailed planning, before you begin to build your solution, helps you to achieve a design that remains efficient for many years to come.

13.2 Best practices

In this section, we present the best practices that we collected from the OnDemand development team and practitioners in the field in how to best design and configure OnDemand applications. We intend to help you to implement archive solutions in the most efficient way.

We include the following best practice topics:

- ▶ Including a Date field in an application group
- ▶ Including a Load Date field in an application group
- ▶ Including an Application ID Field in an application group
- ▶ Application group name handling

- ▶ PDF document access
- ▶ PDF document indexing

13.2.1 Including a Date field in an application group

Most of the time, a document includes at least one date. It is required from a user's point of view for organizing the document filing, although it might not have anything to do with electronic archiving.

For example, an Invoice Number and Customer Number fields provide important information. Without them, we cannot associate the right invoice with the right customer. A date field, such as an Invoice Date, is also needed so we know when this invoice is generated. This information, as well as other date fields, such as Order Date and Delivery Date, are necessary for efficiently keeping documents organized.

Similarly, with OnDemand, to optimize its internal organization and ensure efficient document search and retrieval tasks, we recommend that you include a date field in an application group as a segment. See Figure 13-1 for an example.

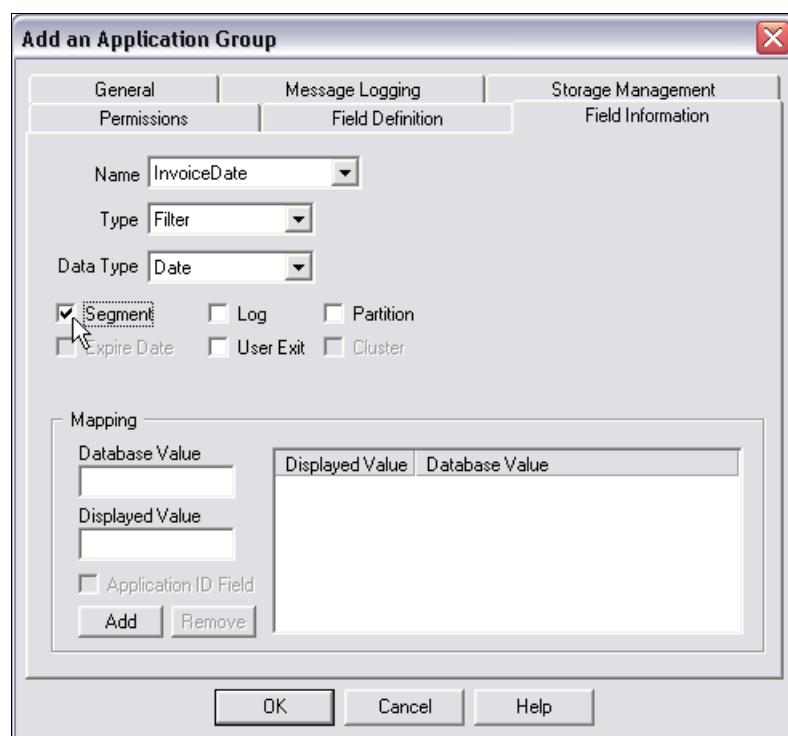


Figure 13-1 *InvoiceDate as segment date*

You should identify a date field that OnDemand can use to segment the application group index data. The segment field enables the searching of specific tables of application group data rather than all of the tables.

In case no date field was chosen as a segment, whenever you select another tab or select OK for the Application Group creation, OnDemand reminds you that it is advisable to do so. See Figure 13-2.

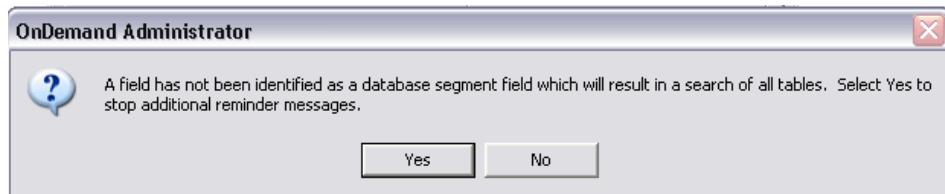


Figure 13-2 Message indicating no date field chosen as a segment

If you see a message like the example in Figure 13-2, select **No** and choose a date field as a segment.

Important: After the application group is added, it is not possible to choose a date field as a segment.

In case no date is available in the document, use at least the Load Date as the segment. See the next section for more information.

13.2.2 Including a Load Date field in an application group

For the IT team, the application dates, such as Invoice Date or Delivery Date, might not be important. Their job is to archive documents and ensure that all the archived documents will be available through OnDemand. Indexing documents with the Load Date field is an efficient way for the IT team to keep track of the archiving activity.

The Load Date of the document might be different from any of the application dates. For example, the invoices can be loaded the day that they are printed or some days later. In this case, the Load Date is different from the invoice Date. Accurate and easily accessible Load Date information helps to avoid any misunderstandings.

In addition to help keep track of archiving activity, the availability of a Load Date index might be of great help in case of an audit or compliance request.

Sometimes, a document does not have any date. In this case, it is useful to use Load Date as a segment date as follows:

1. Define a LoadDate in the application group, in addition to other fields.
2. Within the application definition, click the **Load Information** tab.
3. On the Load Information tab, complete these tasks:
 - a. From the Application Group DB Name, select **LoadDate**.
 - b. In the Default Value field, type the letter t (in lowercase). This triggers OnDemand to store the date, on which the input is loaded into the system, in the Database field. See Figure 13-3.

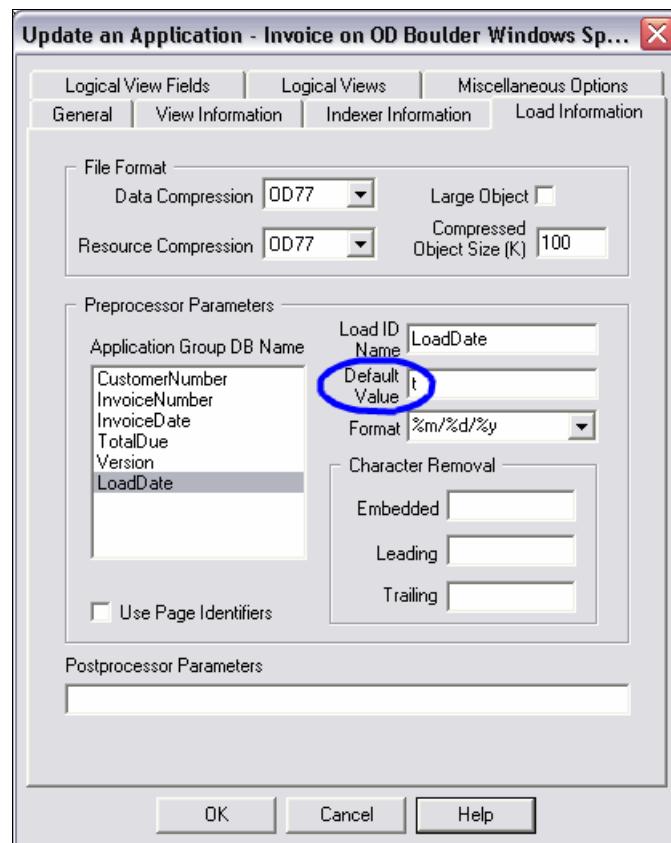


Figure 13-3 Specifying a load date

13.2.3 Including an Application ID Field in an application group

Sometimes, a document layout might change. Even if the indexes stay the same, you might have to add a new application to take into account the new positions of the fields in the document. The updated document might also be produced in another data stream.

To create the new application, use the following steps:

1. Create a new application group by copying the one used for the actual archives.
2. Create a new application, and assign it to the newly created application group.
3. Add the new application group to the folder or folders used to access the documents.

You can also set this up, by using *version support*, as explained in the following steps:

1. Whenever you add a new application group, define a Version field in addition to the other fields that you need for the document. Figure 13-4 shows an example of the Version field.

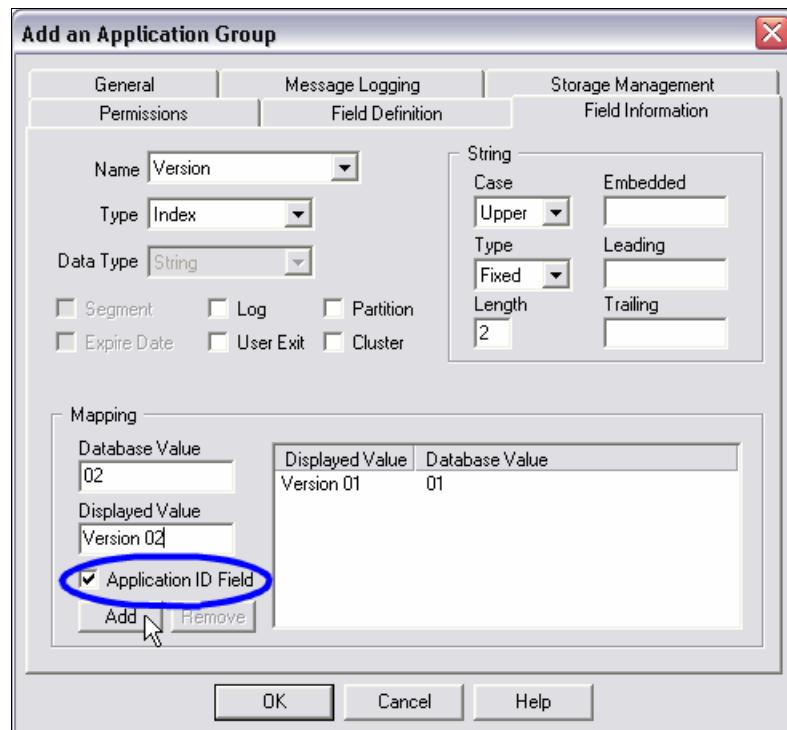


Figure 13-4 Application ID Field

Set up the Version field as follows:

- The field must be of String Data Type.
- A Length of 2 allows you to define 99 versions using only numeric characters. There is no need to specify a greater value. The impact on index database volume is low.
- Application ID Field is selected.
- Mapping values are added for the first version. It is possible to add new values for subsequent versions of the document.

2. Whenever you add an application to the application group, select the version ID, that is the Application ID Field. See Figure 13-5.

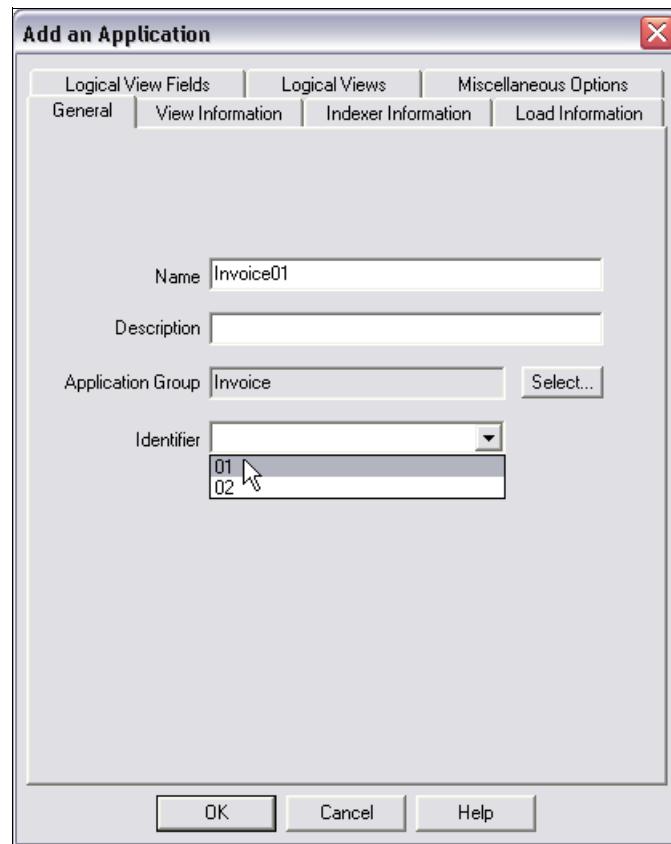


Figure 13-5 Adding an Application with a version ID

This setup offers the following advantages:

- There are structured links between one application group and the applications.
- There is no need to add a new application group to a folder or folders.
- It offers database access optimization because the folder will continue to access only one database, independently of the number of versions.

If a few of applications are linked to the same application group, the application group name and application name must be specified for the load:

- ▶ If you run **arsload** as a daemon or a service, then one of the following actions may occur:
 - The input file name must consider the application to be used.
 - The -A parameter of **arsload** has to specify the part of the file name that identifies the application to load, MVS, JOBNAM, DATASET or FORM.
MVS.JOBNAME.DATASET.FORM.YYYYDDD.HHMMSS.TARD
- ▶ If you run **arsload** as a command line, the -A parameter must specify the application name.

13.2.4 Application group name handling

A *folder* provides a user a way to query and retrieve data stored in OnDemand. A folder provides users with a convenient way to find related information stored in OnDemand, regardless of the source of the information and how the data was prepared.

A folder allows an administrator to set up a common query screen for several application groups that might use different indexing schemes, so that a user can retrieve the data with a single query. For example, you can set up a folder called “Customer Information” that contains orders and invoices from information stored in different application groups, defined in different applications, and created by different programs.

Users can have access to different document types through one folder. They can limit their search to a specific document type, or they can see the document type that each hit-list entry represents.

You can set this capability by defining two additional fields in the folder:

- ▶ To display the application name, define a Document type field as shown in Figure 13-6.

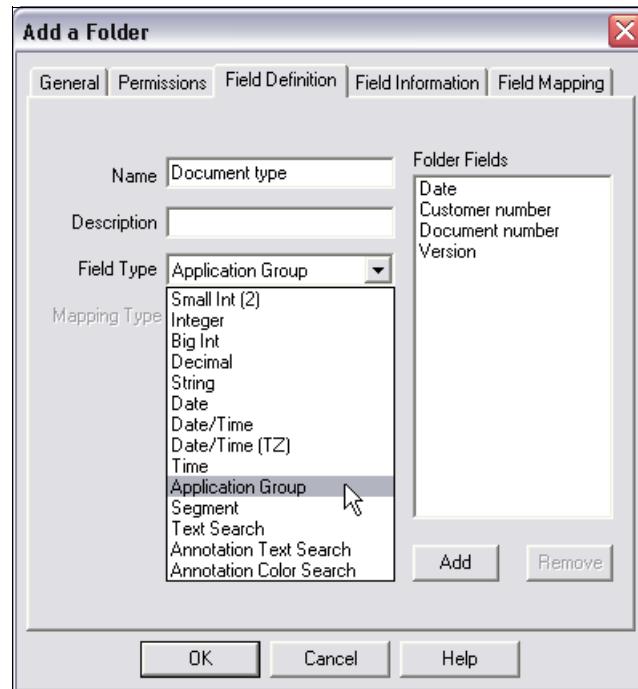


Figure 13-6 Defining an Application Group field

Note the following points:

- This field does not have to be mapped.
- This field is of the *Application Group* field type.

- This field is displayed within the Search Criteria part and Document List part of the OnDemand client. See Figure 13-7.

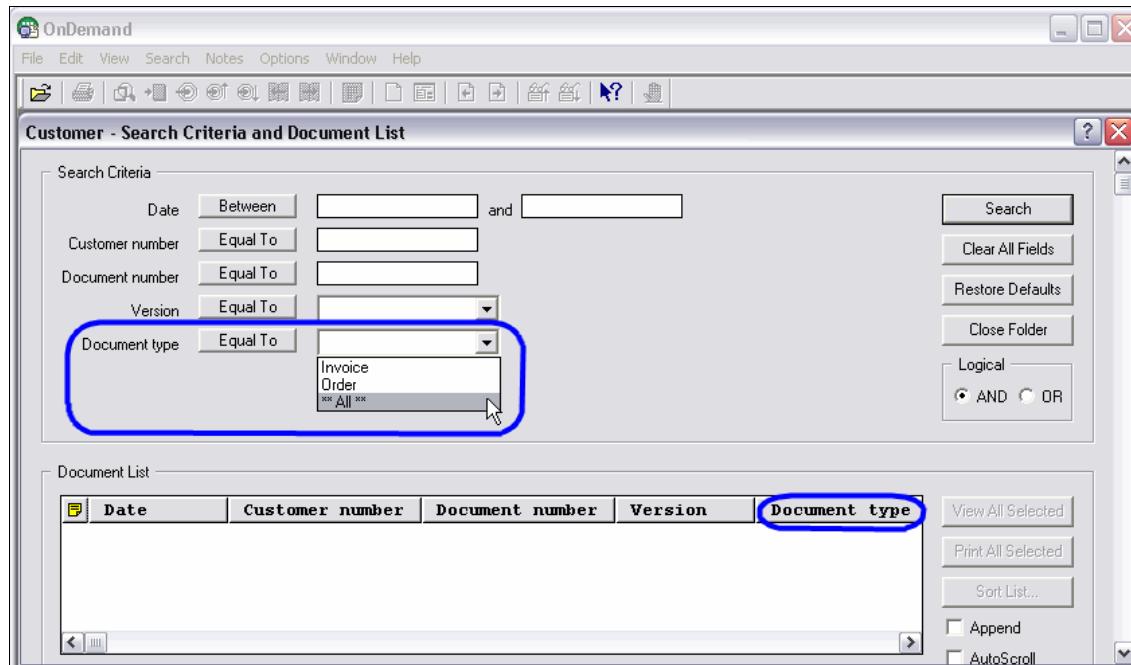


Figure 13-7 The Application Group name displayed in OD client

- If you assign multiple applications to the same application group, you can display the Application ID field; refer to 13.2.3, “Including an Application ID Field in an application group” on page 443:
 - Define a field in the folder.
 - Map this field with the corresponding Application Group field. This is the Version field in the example of the referenced section.

The information is shown the same way as for the Application Group field named Document type. See Figure 13-7.

The information coming from the application groups is displayed to users through a folder. Remember to use self-explanatory and user-friendly expressions for them.

Note: An application group name can be updated after the application group is added, as long as the Application ID field value has not been used as the identifier in an application; otherwise, you can no longer update the application group name. Figure 13-8 shows for the error message that is displayed.

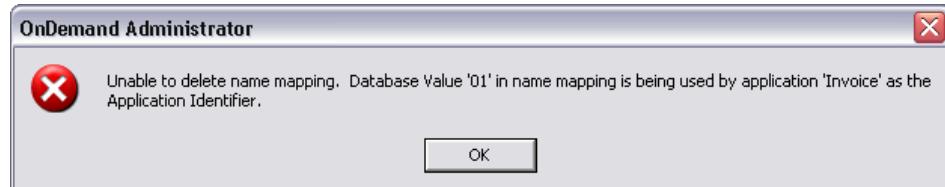


Figure 13-8 Error message: Application ID value used as identifier in application

13.2.5 PDF document access

When you add an application for the archiving PDF documents, there are two tabs in the application definition window that you must set up:

- ▶ The *Indexer information tab*, where you specify how OnDemand indexes the documents (see Figure 13-9):
 - **PDF:** Set the OnDemand PDF Indexer to analyze data stream, segment data, and extract the indexes.
 - **Generic:** You must provide the generic indexer file with all the information needed by OnDemand to archive the file.

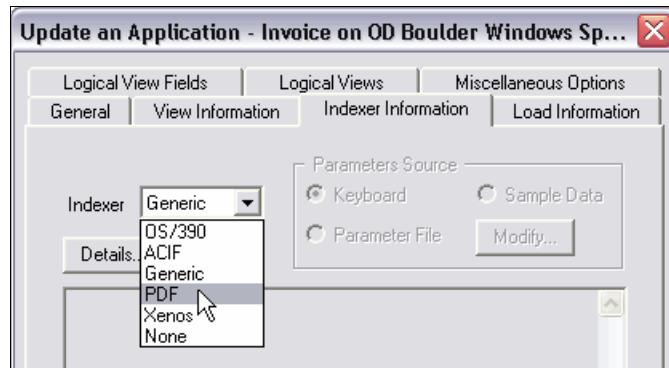


Figure 13-9 Indexer Information tab

- The *View Information tab*, where you specify the way OnDemand displays the PDF documents

They are two ways to support the PDF documents (see Figure 13-10):

- **PDF:** This option provides a PDF seamless integration in OnDemand. This selection supports annotation in OnDemand client and provide full text search support.

Attention: The chargeable Adobe Acrobat product has to be installed on all workstations with OnDemand Client.

- **User Defined with File Extension PDF:** An Acrobat Reader window outside of OnDemand client displays the PDF documents. There is no annotation support in OnDemand client, and the selection does not provide full text search support.

The free Acrobat Reader product must be installed on all workstations with OnDemand client to use this option.

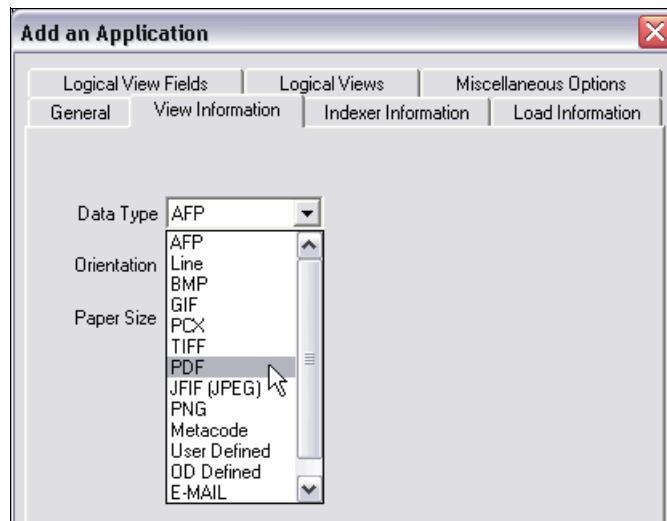


Figure 13-10 View Information tab

Before you decide which way you want to set up OnDemand application for PDF, ask yourself these questions:

- Will the OnDemand clients be used to access PDF documents? Is seamless integration mandatory for your business requirements?
- Is full text search support mandatory?

Important: After you add the application, you *cannot* change the set up in the View Information tab. You must make the right choice in the beginning.

13.2.6 PDF document indexing

When you index the PDF files, sometimes the right information in a PDF document is not captured. One of the following reasons might cause the problem:

- ▶ The different pieces of information in the PDF document are too close to each other and are written using a small font. Therefore, it is not possible to define an efficient area with the graphical indexer or by using the `arspdump` command to capture the right information.
- ▶ The information is printed on all the pages except the first one in the document. For example, you want to index invoices by the total amount due that is printed on the last page of each invoice, and the invoice might have more than one page. A trigger float does not exist for the PDF files as it does for the line-data or AFP files.

If you can change the way how the PDF documents are generated, you can help to solve these problems.

The best context for an efficient index capture is to have:

- ▶ All the necessary information printed on the first page of each document
- ▶ All information printed with the same font at a predefined area

If you cannot change the layout of the PDF documents for business reasons, try the following steps:

1. Add all the required information in a blank part of the first page of the document by using a fixed font and separating clearly all the different pieces of information.
2. Define the PDF indexer parameters using the graphical indexer.
3. Test and validate the indexing.
4. Turn the color of the added information to white so that it does not appear on the printout.



Troubleshooting

The purpose of this chapter is to help the OnDemand system administrator with troubleshooting common OnDemand problems. In this chapter, we introduce the new trace facility that is available in OnDemand for Multiplatforms.

In addition, we cover the following topics:

- ▶ Troubleshooting FAQ
- ▶ Information collection
- ▶ OnDemand trace facility

14.1 Troubleshooting FAQ

This section contains frequently asked questions by the OnDemand administrators. It also includes solutions to common problems encountered by the OnDemand administrators and users.

14.1.1 Determining the nature of the problem

There are several main areas where problems can occur. We classify them into the following categories:

- ▶ Indexing or loading issue
- ▶ OnDemand maintenance issue
- ▶ OnDemand startup problem: **arssockd**
- ▶ OnDemand client issue
- ▶ OnDemand Web Enablement Kit (ODWEK) matter

Tip: For the UNIX platform, the console message might help to determine the cause of the problem. However, if you use Telnet from your PC, you might miss the important console message. For AIX, you can switch the console to your current terminal by using the **swcons ‘tty’** command. To switch it back to the console, simply use the **swcons** command.

In the following sections, we discuss some of these problems that you might have encountered and we provide possible solutions to the problems.

14.1.2 Indexing or loading issue

The following problems are some of those that are encountered while indexing or loading:

- ▶ **Problem:** When you attempt to index a report with a large record length, you see the following error message:

0425-422 AN ERROR OCCURRED WHILE ATTEMPTING TO READ /filename RETURN CODE 310.

Reason or resolution: You might have exceeded the maximum record length for Advanced Function Presentation (AFP) Conversion and Indexing Facility (ACIF), which is 32 K.

- ▶ **Problem:** You add some images, such as a logo, to an AFP document. Subsequently, **arsload** processing slows down by more than 10 times. You also notice that the **arsload** program spends most of the time in the indexing phase. What happened?

Reason or resolution: You added IM image structured fields to the data and ACIF tries to convert them to IOCA. To overcome this, add the ACIF parameter `imageout=asis` to the indexing parameters. ACIF Indexing, and as a result OnDemand loading, run much faster with `imageout=asis`. This is documented in *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235.

- ▶ **Problem:** The **arsload** program is performing progressively slower over time.

Reason or resolution: Performance problems can be caused by a variety of reasons and require careful examination. OnDemand issues an SQL DELETE against the ARSLOAD table before it adds that same information to the ARSLOAD table to guarantee uniqueness. There cannot be duplicated information in the ARSLOAD table. This SQL DELETE is a single action against the ARSLOAD table and uses an index formed from AGID and NAME.

A new index called ARSLOAD_NAME_IDX was added in 7.1.1.0. It contains the AGID and NAME columns for this performance reason. Without this index each load performs a complete table scan of the ARSLOAD table. Upgrading to at least 7.1.1.0 and by issuing the following commands creates the extra index, which might improve the performance.

```
arsdb -efv  
arsdb -rv
```

- ▶ **Problem:** OnDemand does not break up the PDF file into separate reports when TRIGGERS are defined properly and indexing is successful. For some reports, the trigger is not honored and the reports are grouped together.

Reason or resolution: The FIELD value must change for OnDemand to indicate a report break. In Example 14-1, there are several pages of a document; Page 1 is the TRIGGER, and the name is the field that is placed into the index.

Example 14-1 Sample index

Page 1
John Doe

Page 2
John Doe
...
...

```
Page 1  
John Doe  
...  
Page 1  
John Smith
```

In this example, since 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:** You run OnDemand on HP/UX and encounter the error message shown in Example 14-2 while attempting to index a PDF document.

Example 14-2 Error while indexing PDF

```
/usr/lib/dld.sl: Unresolved symbol: AGMDeleteRasterDev (code) from  
/opt/ondemand/lib/libCoolType.sl  
/usr/lib/dld.sl: Unresolved symbol: AGMInit (code) from  
/opt/ondemand/lib/libCoolType.sl  
/usr/lib/dld.sl: Unresolved symbol: AGMNewRasterDev (code) from  
/opt/ondemand/lib/libCoolType.sl
```

Reason or resolution: This is an installation problem with OnDemand and HP/UX. To resolve this problem, list the contents of /opt/ondemand/lib directory by using the following command:

```
ls -l
```

This command produces the output shown in Example 14-3.

Example 14-3 Contents of /opt/ondemand/lib

```
-r-xr-xr-x 1 root root 1762704 Dec 17 2003 libACE.sl  
-r-xr-xr-x 1 root root 9797408 Dec 17 2003 libAGM.sl  
-r-xr-xr-x 1 root root 4431076 Dec 17 2003 libBIB.sl  
-r-xr-xr-x 1 root root 5950980 Dec 17 2003 libCoolType  
-r-xr-xr-x 1 root root 4051004 Feb 21 2001 libCoolType.sl  
-r-xr-xr-x 1 root root 630276 Dec 17 2003 libOPP.sl  
-r-xr-xr-x 1 root root 270944 Dec 12 2001 libodxtra.sl  
-r-xr-xr-x 1 root root 17307844 Dec 17 2003 libpdfl.sl
```

Save the old libCoolType.sl file as libCoolType.sl.orig, and rename the newer libCoolType file to libCoolType.sl to resolve the PDF indexing problem.

- **Problem:** You receive a segmentation fault when loading PDF documents on AIX.

Reason or resolution: Check the PDF version of the document.

- a. Select **File** → **Document Properties**.
- b. In the Document Properties window that opens, select **Description** (see Figure 14-1). In this example, the PDF version is 1.3.

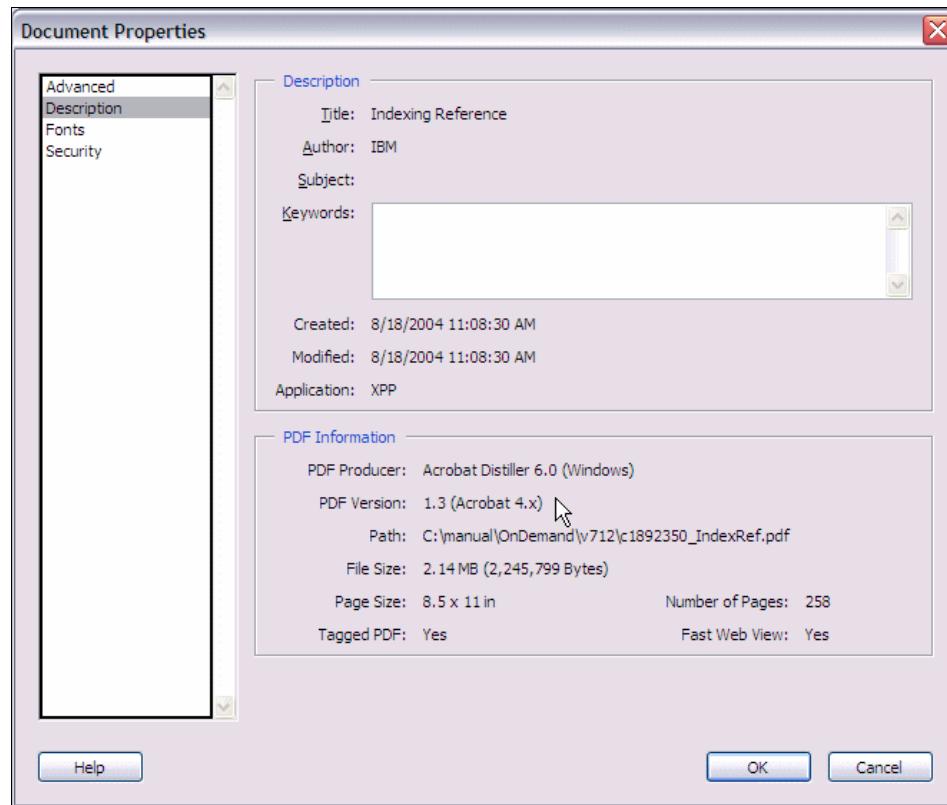


Figure 14-1 Document Properties window for the PDF document

If your PDF version is 1.5 and later, you must upgrade your OnDemand server to the latest version to avoid the segmentation fault. This is because starting at version 7.1.2.5, OnDemand uses new libraries that support the newer PDF versions.

- **Problem:** The OnDemand Windows client hangs when performing a Full Text Search against PDF version 1.5 and later.

Reason or resolution: Similar to the previous problem, you must upgrade the OnDemand server to the latest version to resolve the problem.

14.1.3 OnDemand maintenance

The following problems, among others, are related to OnDemand maintenance:

- ▶ **Problem:** One of the OnDemand database file systems is reaching 100% utilization, and there is no way to increase the file system size. How do you determine if an application group is using this file system?

Reason or resolution:

- a. Use 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 as follows:

```
Table still open for loading: ApplGroup(AppGrpName) Agid(5016) Table  
(CAA1)
```

- b. List the tablespace ID, tablespace, and table name for the application group data table that is opened, for example:

```
su - archive  
db2 connect to archive  
db2 "select tbspcieid, tbspcname, tabname from syscat.tables where  
tabname='CAA1'"
```

The command returns the following with tablespace ID 3:

```
TBSpaceID TBSpace TABNAME  
3 ROOT_CAA1 CAA1
```

- c. Determine the containers for this tablespace ID with this command:

```
db2 "list tablespace containers for 3"
```

The command returns with the tablespace containers for tablespace 3:

```
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
```

- d. Check if any of the containers listed previously belong to the file system, which is full.

- If they do, close the opened application group data table using the following command:

```
arstblsp -a 1 -g AppGrpName
```

The following message indicates that the table has closed successfully.

```
Closed table successfully: App1Group(AppGrpName) Agid(5016)
Table(CAA1)
```

- If they do not, continue to find the next application group.

When the application group data table is closed, OnDemand creates a new table on a file system as defined in ARS.DBFS when data is next loaded. It also searches for the file system with more free space to create the new table.

- **Problem:** The **arsmain** program fails to complete.

Reason or resolution: The common problem encounters when **arsmain** is a full cache file system or broken links.

For a full cache file system, check which file system is full, and expand the file system if possible.

For a broken link problem, the system log displays errors relating to **arsmain**.

If neither situation is the case, check to see if **arsload** is running at the same time. If **arsload** is running at the same time when you run the **arsmain -r** command, **arsmain** might fail.

14.1.4 OnDemand startup problem

The **arssockd** startup problem with the AIX server running OnDemand. When the command **arssockd** is run, there is no error message but nothing is started.

In this case, check the console message of the AIX server for an error message similar to the one shown in Figure 14-2.

```
arssockd (REDBK): 03/14/06 08:21:18 0 ARSSOCKD 2 13 DB
Error: [IBM][CLI Driver] SQL1224N A database agent could not be started to
service a request, or was terminated as a result of a database system
shutdown or a force command.
SQLSTATE=55032 -- SQLSTATE=08001, SQLCODE=-1224, File=arssys.c, Line=367
```

Figure 14-2 Error message from the console

If the console messages match the example in Figure 14-2 on page 459, then enter the following command to turn on the EXTSHM variable:

```
export EXTSHM=ON
```

On the same terminal, try to start **arssockd** again. If it still fails to start, then as the DB2 instance owner, enter the following command:

```
db2set DB2ENVLIST=EXTSHM
```

This command sets EXTSHM for DB2 as well.

14.1.5 OnDemand client issue

Some of the common client issues and problems are as follows:

Tip: If your users have a problem with the OnDemand client and this problem only happens with that particular client, you might save time by first reinstalling the client on that computer. Then run a test to see if the problem still persists after the reinstallation.

- ▶ **Problem:** Does the Content Manager OnDemand Windows client support multiple monitors?
Reason or resolution: There is an issue with using multiple monitors at higher resolutions, for example 3840 x 1024, that can cause the client to crash upon startup. The error occurs in an area of code just before or during thumbnail creation and can be bypassed by disabling the thumbnail feature of the client. To disable the thumbnail feature:
 - a. Add a String Value entry with the name THUMBNAILS to the registry
HKEY_CURRENT_USER → Software → IBM → OnDemand32 → Client → Preferences.
 - b. Set the value of this entry to 0.This does not affect the Content Manager OnDemand Windows Administrative Client.

- ▶ **Problem:** You see an error message indicating that the client is unable to load the ARSUSDOC.DLL module when performing PDF Text Searches using Content Manager OnDemand for Windows.
Reason or resolution: This problem occurs only when you attempt to do a text search on PDF documents. The problem resides in the arsview.exe file not having what it needs on the Windows server to execute properly. One solution is to add the OnDemand client install directory that is on the server to the server PATH environmental directory. This enables arsview.exe to function properly. You must restart the OnDemand Services for the new PATH to become effective.
- ▶ **Problem:** On Windows XP, you receive the “File not found” error message when you attempt to view an Excel® Document using ODWEK.
Reason or resolution: This is due to a change made to the way Internet Explorer communicates to Excel. You can obtain the fix only directly from Microsoft. The Microsoft Knowledge Base Article number is Q888405.

14.1.6 ODWEK matter

In addition to the last problem mentioned in the previous section about ODWEK, you might also need to know how to configure ODWEK to automatically install the SUN Java Plug-in.

You can achieve this by adding the ODApplet.jre.path.IE statement to the arswww.ini file. You can prompt the user as to whether they want to install your version of the Sun Java™ Plug-in. Keep in mind that this statement must end with the version specification; for example the text "#Version=1,4,2,0" or errors can result when attempting to install over older versions of the Sun Java Plug-in.

14.2 Information collection

If the guidance in 14.1, “Troubleshooting FAQ” on page 454, does not help you in determining and resolving your problem, in this section, we explain the information to gather for the IBM Support team to help you more efficiently.

When you report a problem to the support center, first, you must provide the version of the software that you are using. For OnDemand, this might include the operating system, DB2, Oracle, Tivoli Storage Manager and OnDemand, and ODWEK. This information helps the Support Team to determine whether the software version is still supported and whether there are known issues to the software level.

Table 14-1 shows the different commands used to determine the version of OnDemand on different operating systems.

Table 14-1 Determine the version of OnDemand in Multiplatforms

OS	Example of the command to determine the version
IBM AIX	#lslpp -l grep ars ars.srvr 7.1.2.5 COMMITTED IBM DB2 Content Manager ars.www 7.1.2.5 COMMITTED IBM OnDemand Web Enablement
Sun Solaris	#/usr/bin/pkgparam -v ondemand PKGINST NAME VERSION INSTDATE PKGINST='ondemand' NAME='IBM DB2 Content Manager OnDemand for Sun Solaris' VERSION='7.1.2.5.DSP=7.1.2.5'
HP/UX	#swlist -l product grep OnDemand ODWEK 7.1.2-3 IBM OnDemand Web Enablement Kit for HP-UX OnDemand 7.1.1.0 IBM Content Manager OnDemand for HP-UX
LINUX	Look for the highest version for the package name in the list. In the example, the ODWEK Version is 7.1.2.4. # rpm -qa grep odwek The versions are: <ul style="list-style-type: none">▶ odwek_license-7.1.2-0▶ odwek-7.1.2-0▶ odwek_icu-7.1.2-2▶ odwek_icu-7.1.2-4▶ odwek-7.1.2-1▶ odwek-7.1.1-0▶ odwek-7.1.2-2▶ odwek-7.1.2-4▶ odwek_icu-7.1.2-0
Windows	From the OnDemand configurator, click Help → About .

After you get the correct version number of the software that you are using, you must collect the information specific to the problem.

There are several main areas where problems can occur. We divide them into the following areas in this section:

- ▶ Indexing or loading
- ▶ Database
- ▶ Tivoli Storage Manager
- ▶ OnDemand client logon
- ▶ Performance
- ▶ ODWEK
- ▶ OnDemand server hang or crash

In 14.2.8, “Exporting information to a local server” on page 469, we demonstrate how to export OnDemand information, such as an application group, application, and folder, into local server.

14.2.1 Indexing or loading

This section describes the logs to be collected that relate to indexing or a loading problem.

Common load issue

Table 14-2 shows the information to collect if there be a problem with loading.

Table 14-2 Information to collect for loading

File name	Description
ARSSOCKD.ERR	This is the log file for <code>arssockd</code> daemon process. The process is instance dependent if multiple instances are running.
ARSLOAD error message	The ARSLOAD error message shows whether it failed at the indexing or loading phase.
ARS.INI	This is the OnDemand instance configuration file. Each instance has a section in the ARS.INI file.
OnDemand System log	This is the OnDemand system logs in system log folder. There are various message number regarding warnings or errors at the time of failure.
Export of 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 dump generated by the operating system.
Version or Level of DB2/Oracle/SQL server and OnDemand	This file name contains the version or level of software that the server is currently using. Sometimes a problem might be resolved by upgrading to the latest PTF.

The manual *IBM Content Manager OnDemand - Messages and Codes*, SC27-1379, contains the error message codes from OnDemand system log.

Common AFP indexing problem

OnDemand cannot load AFP data without indexes; therefore, you must first make sure that your AFP data is already indexed. This means that the AFP must have TLEs.

Table 14-3 shows the information to collect when you have problems with AFP.

Table 14-3 Information to collect for common AFP problems

File name	Description
Export of Folder, Application Group and Application files and sample data	The export files are used to import to the test server for problem replication.
ACIF indexer error message	This file contains the error messages generated by the ACIF indexer.
AFP sample data file	This should be a non-confidential data file that can be viewed by the support team to verify AFP syntax.
AFP interim files used by AFP viewer within OnDemand Windows Client	The files are created in the OnDemand client directory under C:\ProgramFiles\IBM\OnDemand32\DATA. These files are deleted automatically after the document is closed by viewer. The file is useful in determining whether it is a server or client issue.
AFP trace report	AFP trace can be turned on by modifying the FTDPOR2.INI file in the OnDemand client directory, C:\Program Files\IBM\OnDemand32.
AFP resource and font files	Sometimes this file is useful for various AFP issues such as overlay, company logo, or national language support (NLS) fonts.

Before you log a problem to the support team, use the information in Table 14-3 to look for clues for your problem. Especially regarding the error codes from the ACIF indexer, you can check the error codes in the manual *IBM Content Manager OnDemand - Messages and Codes*, SC27-1379. You might find the solution right away. If you have an AFP dump tool, you can also dump the AFP data file to check for 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 OnDemand. The loading of AFP data requires more specific AFP structure than printing. The manual *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235, provides information about the correct AFP data stream structure.

14.2.2 Database

For DB2 problems, collect the information in Table 14-4 for problem determination.

Table 14-4 Information to collect for DB2

File name	Description
db2diag.log	This is the DB2 diagnosis file. It is located in the \$HOME/sqllib/db2dump directory, where \$HOME is the home directory of the DB2 instance.
CLI trace	This file contains the call level interface (CLI) trace file for diagnosis SQL statements. The CLI trace option must be turned on to collect the file.
SQLCODE error message	If it is available, collect this information to determine whether the problem is from OnDemand or the database. See the example in Figure 14-2; the SQL error code is 1224.
DB2 configuration report of OnDemand instance	This report is generated by the db2 command: <code>db2 get db cfg for instance_name</code>
Application Group Report	The Application Group ID is the name of the respective DB2 tables.

Setting the CLI trace for DB2

We list two methods to turn on the CLI trace for DB2. One method is to do direct editing of the db2cli.ini file. The other method is to use the DB2 command line.

The examples shows the common option for the DB2 CLI trace. The support team might have a different option to collect information as appropriate to your situation. Modify these options as advised.

In both cases, the trace file to be collected is /tmp/db2trace.dmp.

Method 1: Setting up the trace by editing the db2cli.ini file

You can set up trace by editing the db2cli.ini file as follows:

1. Add a section similar to the one in Example 14-4 in the db2cli.ini file.

For Windows, this file is present in the \sqlib path, for example, C:\Program Files\IBMs\SQLLIB. For the UNIX platform, it is placed in the /sqlib/cfg path of the home directory of the instance owner, such as /home/archive/sqlib/cfg.

Example 14-4 Common section of the db2cli.ini file

```
[COMMON]
TRACE=1
TRACEREFRESHINTERVAL=5
TRACEFILENAME=/tmp/db2trace.dmp
TRACEFLUSH=1
TRACECOMM=1
```

The full path of the TRACEFILENAME should be a valid directory with permission for everybody to write.

2. Restart the application, in this case **arssockd**, for the changes to take effect.
3. Simulate the DB2 problem that you have encountered to capture the trace information.
4. To turn it off, update the db2cli.ini file again and set TRACE=0.
5. Restart **arssockd** to take effect.

Method 2: Setting up the trace by using the DB2 command line

Alternatively, you can use the DB2 command line to activate the trace:

1. With the DB2 instance, run the DB2 commands as shown in Example 14-5.

Example 14-5 Turning the trace on via 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. Simulate the DB2 problem that you have encountered to 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** to take effect.

14.2.3 Tivoli Storage Manager

For Tivoli Storage Manager-related OnDemand problems, collect the information shown in Table 14-5.

Table 14-5 Information to collect for Tivoli Storage Manager

File name	Description
Application Group Report	The Summary information for Storage Management shows the Storage Set name, which is related to Tivoli Storage Manager.
Storage Set Report	This information provides the node name at Tivoli Storage Manager.
TSM activity log	This log shows the events in the Tivoli Storage Manager server. You can retrieve the log by using the Query actlog command.
TSM error message	Tivoli Storage Manager error messages are prefixed with ANS, ANR, and so on. This error is generated by Tivoli Storage Manager 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 choosing **Summarize**.

14.2.4 OnDemand client logon

If an OnDemand client fails to logon to the server, first check that **arssockd** is running on the server. Second, check the network connectivity by performing a ping test from the DOS windows of the client. Open the DOS windows and **ping** the host name or the IP address of the OnDemand server.

Collect the files listed in Table 14-6 for client problems such as logging into OnDemand.

Table 14-6 Information to collect for client logon problems

File name	Description
ARS.INI	This is the OnDemand instance configuration file. The instance is configured in each section in the ARS.INI file.
ARS.CFG	This is the OnDemand configuration file.
ARSSOCKD.ERR	This is the log file for the arssockd daemon process. The process is instance dependent if multiple instances are running. This file is located in the path defined for ARS_TMP.

14.2.5 Performance

Table 14-7 lists reports and logs that can be used to analyze performance issue.

Table 14-7 Information to collect for performance issues

File name	Description
Application group report	It is useful to check those fields in the report whether they are indexed or filters. Simply reviewing this report might resolve the issue.
Database reorganize information	This file is used to check if the arsdb command has been run to reorganize OnDemand system and data tables.
Memory information	This file contains the amount of physical memory, and the memory setting in the server such as output from the ulimit command.
ARSSOCKD.ERR	This is the log file for the arssockd daemon process. The process is instance dependent if multiple instances are running. This file is located in the path defined for ARS_TMP.
Indexer information from application report	This file helps to determine if the report has a single index, which uses up memory if the report is huge. Also for a large report without using large object option, the client experiences a long time to download.

14.2.6 ODWEK

For ODWEK problems, gather the information as shown in Table 14-8.

Depending on the environment and the specific failure, some of the information might not be present in your environment.

Table 14-8 Information to collect for ODWEK

File name	Description
ARSWWW.INI	This is the ODWEK configuration file.
arswww.log	This is the ODWEK log file. You have to turn on debug mode and restart the Web server for changes to take effect.
httpd.conf	This is the IBM HTTP server configuration file.
was.conf	This is the WebSphere Application Server configuration file.
HTTPD log	This is the IBM HTTP server log file.
OnDemand system log	This file contains the OnDemand system logs from the System log folder.

File name	Description
core	This is the core file generated by the operating system.
Screen shots of the problem	This file contains screen captures of the error message or document. Sometimes it is useful for non-English error message and document.
Plug-ins or applets information	This file helps to check the version in use. Sometimes a problem is resolved by using the latest version.
Version or Level of ODWEK, DB2/Oracle/SQL server and OnDemand	This file indicates the version or level of software that the server is currently using. Sometimes a problem might be resolved by simply upgrading to the latest PTF.

14.2.7 OnDemand server hang or crash

For OnDemand server hang or crash problems, there are a few *MustGather* Technotes that you can search for by going to the following Web address:

<http://www.ibm.com/software/data/ondemand/mp/support.html>

Search this Web site using the keyword `mustgather` to find the following Technotes:

- ▶ MustGather: Content Manager OnDemand Server for Windows - Hang, reference #1223907
- ▶ MustGather: Content Manager OnDemand Server for Windows - Crash, reference #1226443
- ▶ MustGather: IBM DB2 Content Manager OnDemand server hang on AIX, reference #1222374
- ▶ MustGather: IBM DB2 Content Manager OnDemand server crash on AIX, reference #1223109

Follow the instructions from the Technotes to gather information when the server hangs or crashes.

14.2.8 Exporting information to a local server

The support team might require information of the OnDemand application group, application, and folder for problem determination. This section explains how to create a local server to export object information.

1. Create a local server on your PC.
 - a. As shown in Figure 14-3, from your OnDemand administrative client, highlight **OnDemand Servers** and select **File → New Server**.

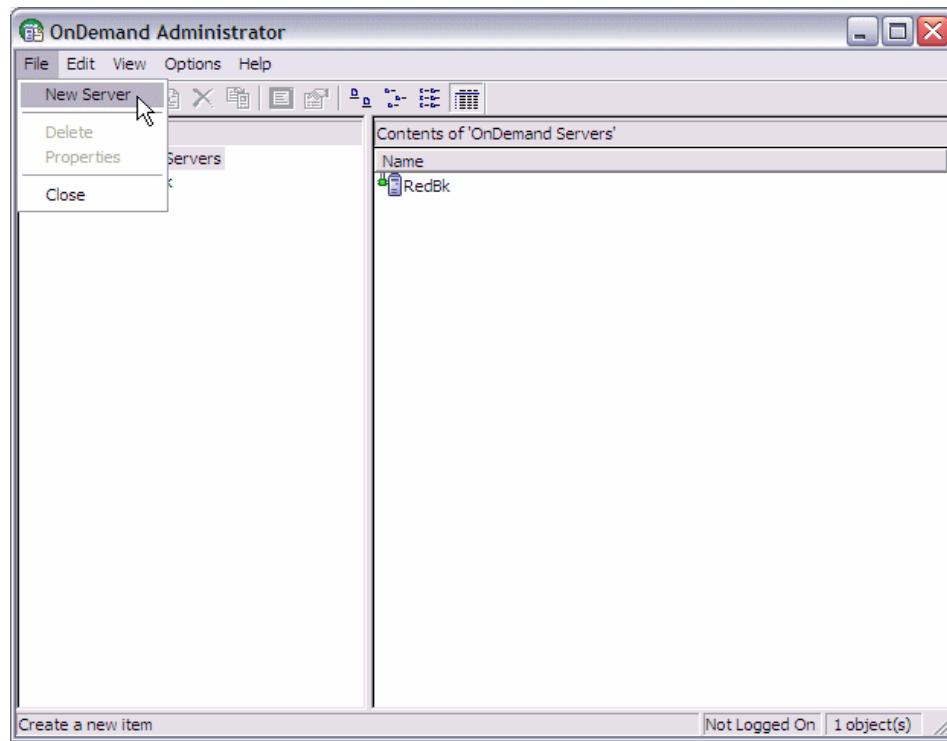


Figure 14-3 Setting up the local server

- b. In the Add a Server window that opens, for Protocol, select **Local**, and enter the information as shown in Figure 14-4. Then click **OK**. A local server with the name **ODlocal** is created.

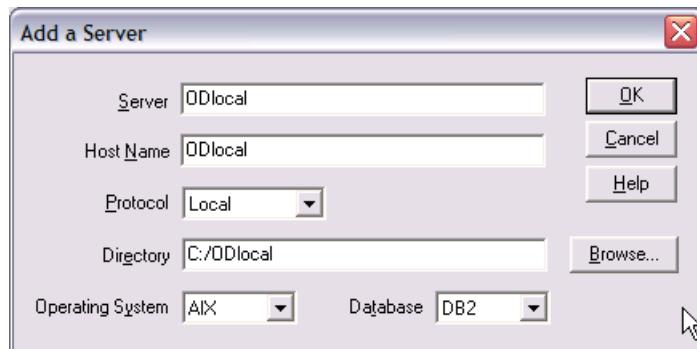


Figure 14-4 Add a Server window

2. The local server cannot be used until it is setup. Right-click the new server **ODlocal** and select **Setup** as shown in Figure 14-5.

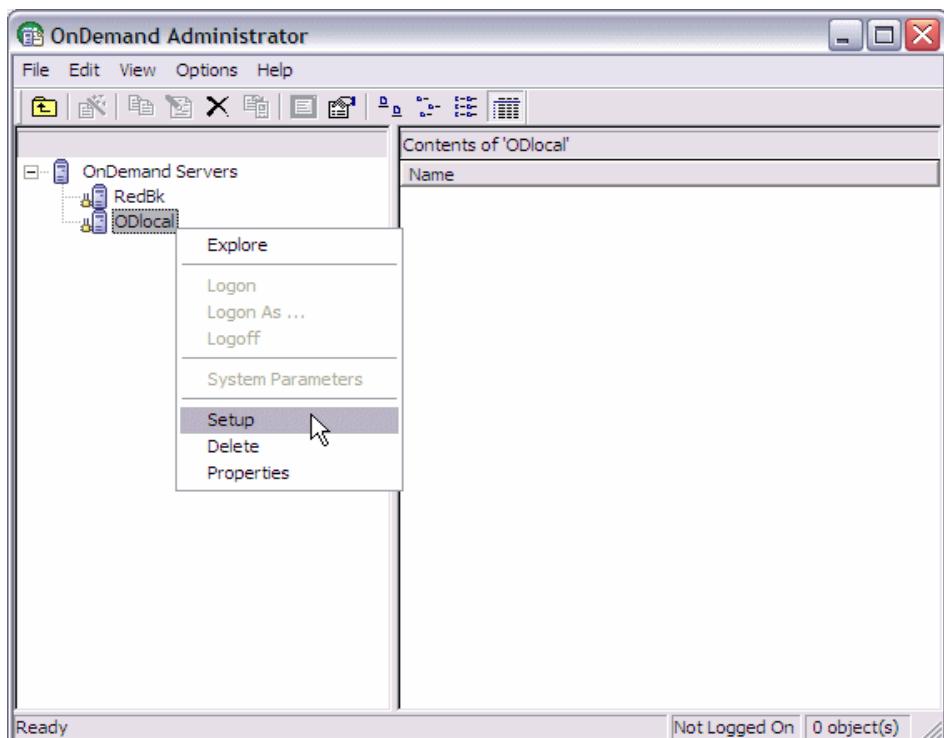


Figure 14-5 Setting up the local server

In the Setup a Local Server. Are you sure? window that opens, click **OK**.

When setup is done, the local server is ready to use. By default, the local server has a user with the name *admin* without any password.

3. Export the information requested 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 14-6.

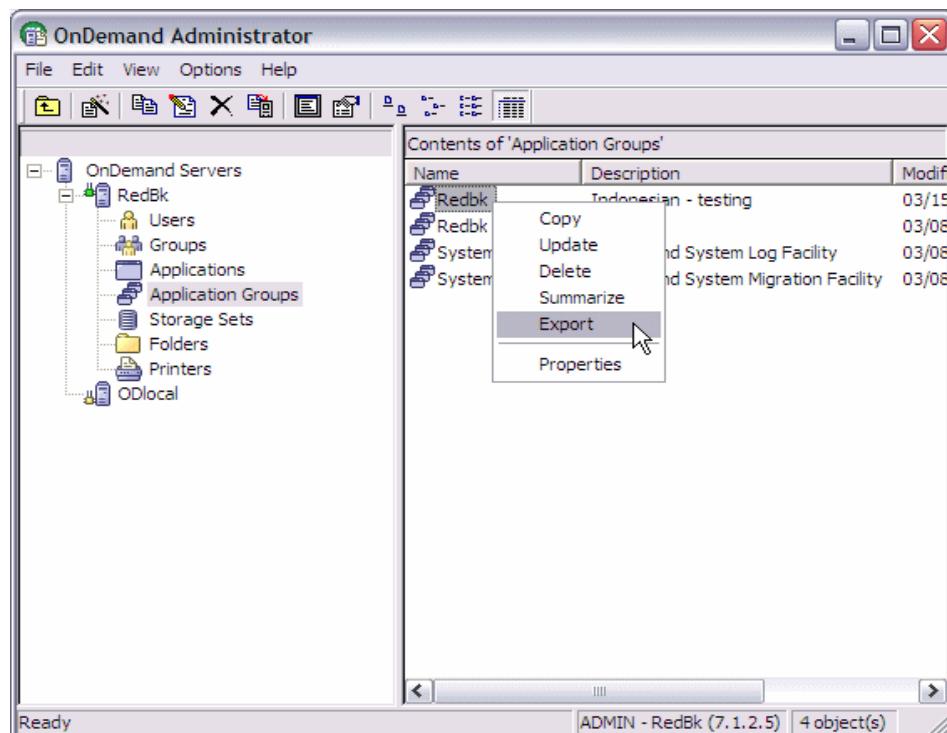


Figure 14-6 Exporting an application group

4. In the Export Application Groups window (Figure 14-7 on page 473) that opens, complete these tasks:
 - a. From the Server list, select the server to be exported.
 - b. Click **Export**. The information of the application group that you have chosen starts transferring to ODlocal.
 - c. Check the message at the end of the export to make sure that it is successful.

d. You can select **Ignore Warnings** or **No Storage Set**.

- Select **Ignore Warnings** if you want OnDemand to add an item regardless of any warnings encountered. Otherwise, OnDemand stops transferring the item when the first warning is encountered. For example, if the application group has users and groups permission 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 already exists on the destination server, the exports also fails.
- Select **No Storage Set** if you do not want OnDemand to assign a storage set to the application group.

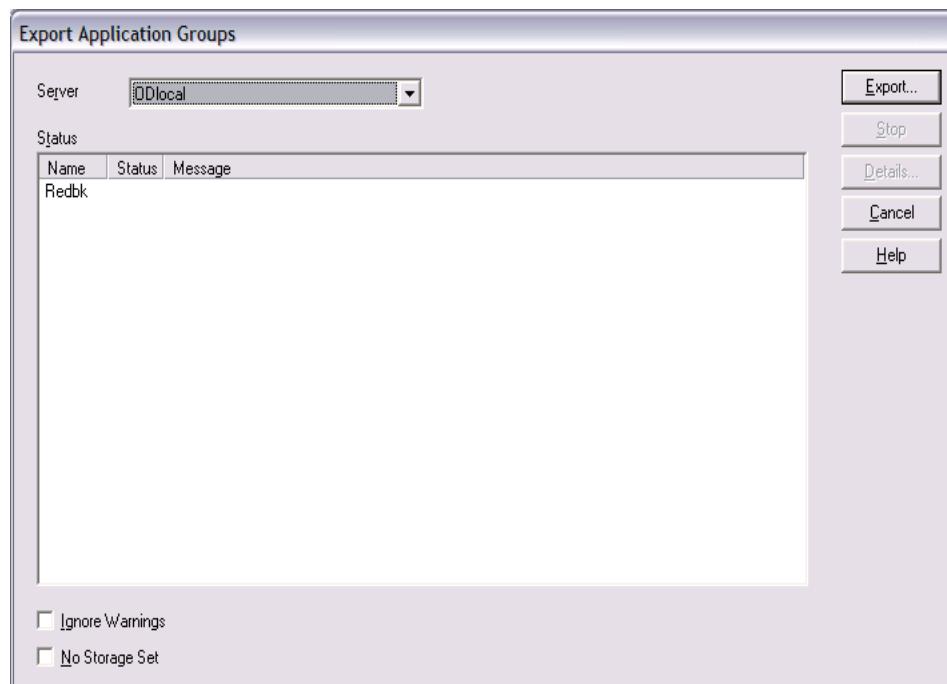


Figure 14-7 Export Application Group window

5. When all the requested information has been exported to the local server, zip the entire directory as defined from the Directory of the local server. In this example, it is C:\ODlocal as shown in Figure 14-4 on page 471.

Tip: When exporting, we recommend this order: printers, users, groups, storage sets, application groups, applications, and folder.

14.3 OnDemand trace facility

OnDemand has incorporated a trace facility into the code to help the support team to perform problem determination. The trace facility is available in OnDemand for Multiplatforms. In this section, we show you how to enable trace.

14.3.1 Enabling the trace facility

To enable the trace facility on the UNIX platform, you must add the following line in the ARS.CFG configuration file.

```
ARS_TRACE_SETTINGS=<path to the trace.settings>
```

For example in AIX, you set following line:

```
ARS_TRACE_SETTINGS=/usr/lpp/ars/config/trace.settings
```

If **arssockd** cannot be started after you set up the trace, you must turn on EXTSHM. Refer to 14.1.4, “OnDemand startup problem” on page 459, for the steps to turn it on.

For the Windows platform, the trace facility is enabled via the OnDemand configurator.

The parameter in the trace.settings file is read when the server starts. It provides the server startup program with the trace options. The OnDemand installation comes with a default trace.settings file for UNIX as shown in Example 14-6. You may modify the file for different options.

Example 14-6 The default trace.settings file

```
#####
#      trace settings - OnDemand Trace Settings File      #
#
#  5622-662 (C) COPYRIGHT IBM CORPORATION 2001          #
#  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 licensee of the OnDemand product is free to copy, revise,  #
#  modify, and make derivative works of this program sample    #
#  as they see fit.                                       #
#
#  File Format:                                         #
#      1) Comments must begin with a # in the first column  #
#      2) Comments cannot exist on the same line as a PARM    #
#
```

```

#      3) PARM=VALUE, no spaces before PARM, no spaces after VALUE,      #
#          and no spaces before/after the equal sign.                      #
#      4) Blank lines are ignored.                                       #
#
# NOTE: Please see documentation for configuring these parameters.   #
#
#   1 ODWEK           ODWEK
#   2 Client-Server   CS
#   3 Cache            CACHE
#   4 Common           CSV
#   5 Communication    XPORT
#   6 Compression     COMP
#   7 Configuration    CFG
#   8 Conversion       XDR
#   9 Database          DB
#  10 Date             DATE
#  11 Iconv            ICONV
#  12 ICU              ICU
#  13 Load             LOAD
#  14 Memory           MEM
#  15 Operating System OS
#  16 PDF               PDF
#  17 Profile          PROF
#  18 Security          SEC
#  19 Server            SRVR
#  20 Network           COM
#  21 Storage Manager   SM
#  22 TSM               TSM
#
#      1      2      3      4      5
# 12345678901234567890123456789012345678901234567890
# OCCCXCCDDIILMOPPSSCST
# DASSPOFDBACCOESDREROMS
# WC VOMGR TOUAM FOCVM M
# EH RP EN D F R
# KE T V
#
# "000000000000000000007000"
#
# Message Levels (Add together, in HEX, to combine levels)
#
# ERROR      0x01
# WARNING    0x02
# INFO       0x04
# FLOW        0x08
#
# By default the SRVR component will have ERROR, WARNING, and INFO
#
#####

```

```
#  
# Parameter overrides for trace output  
#  
#  
#####  
# TRACE_FILE      Will be created in the ARS_TMP directory  
# TRACE_FORMAT    TEXT and CSV are the only outputs currently supported  
# APPEND          0 create new file or 1 append to existing trace file  
# CPU_TIME        Show raw clock timer output (Windows only)  
#####  
  
[TRACE]  
COMPONENT_LEVEL=000000000000000000000000000000007000  
TRACE_FILE=ARCHIVE.trace.log  
TRACE_FORMAT=TEXT  
APPEND=0  
CPU_TIME=0
```

The default trace.settings file has the following component level:

Using information from Example 14-6 on page 474, the nineteenth bit of COMPONENT_LEVEL corresponds to SRVR, which is for server trace. The value 0x07 is a summation of 01 + 02 + 04, which means that the message level of the trace is ERROR + WARNING + FLOW.

Note: Choosing the message level FLOW or INFORMATION might result in excessive log information.

Although the value of the TRACE_FILE can be changed to any name, we recommend that the name of the TRACE_FILE relates to the instance name for easy identification. The default value for the TRACE_FILE is ARCHIVE.trace.log. This file is created in the directory path of ARS_TMP.

For multiple instances, you may specify a different file name and path for ARS_TRACE_SETTINGS in the ARS.CFG file of that instance. Then in the trace settings file, you may specify a unique name for the TRACE_FILE.

Note: You must restart `arssockd` for OnDemand to read in the trace settings from the configuration files.

The previous trace settings are useful when you cannot activate **arssockd**. Next we look at traces that can be started by the OnDemand administrative client.

14.3.2 Setting trace parameters in the OnDemand administrative client

After you enable tracing, you might set the appropriate option for a runtime trace via the OnDemand administrative client.

After you log on to the OnDemand administrative client, you can configure tracing as explained in the following steps:

1. Right-click the **server name** and select **Trace Parameters**. Figure 14-8 shows how to enable tracing from the OnDemand administrative client.

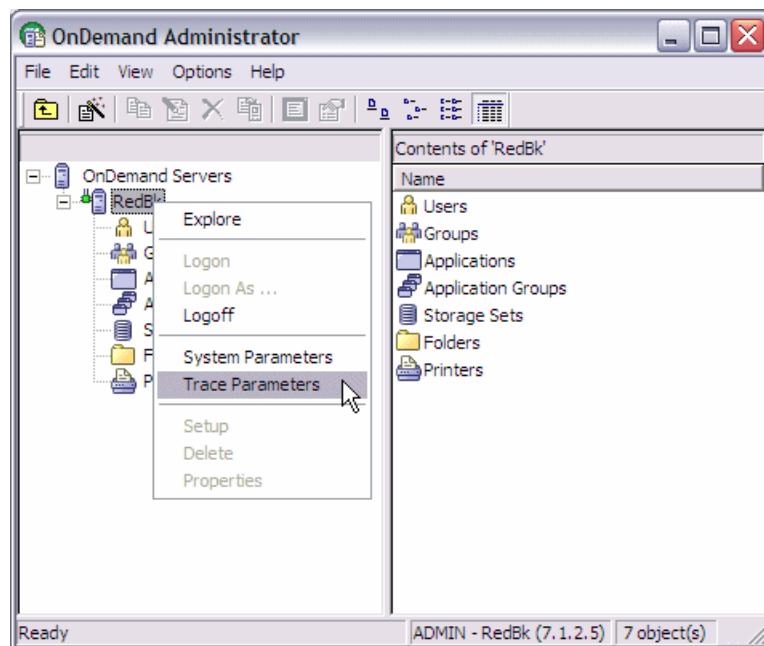


Figure 14-8 Enabling Trace Parameters from the administrative client

2. In the System Trace Setting window (Figure 14-9 on page 478), complete the following steps:
 - a. Select **Activate System Trace** to enable tracing on the server.
 - b. In the Components To Trace section, click the component name to select the component that you want to trace.
 - c. In the Trace Level Reporting section, you might also set the message level of the trace for each components. The values provided for the message level is similar to the COMPONENT_LEVEL in the file trace.settings. For problem determination, consult your IBM support team on the appropriate trace to capture.

- d. Click **Update** to make your choice effective; you do not need to restart OnDemand.

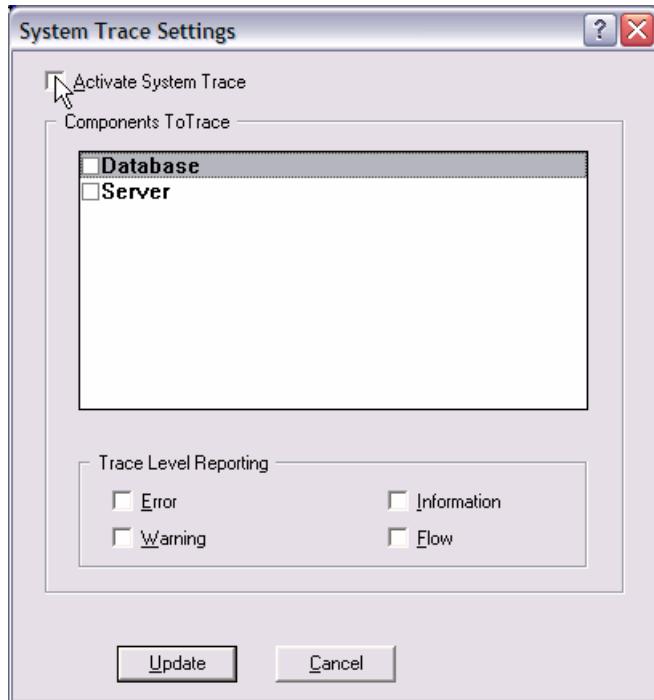


Figure 14-9 System Trace Settings window

Note: You can stop or start the runtime trace from the administrative client anytime without restarting `arssockd`.

After the trace is collected, you can send the trace file to the IBM Support team.

Important: Only use trace with the help of IBM Support since activating it might severely impact the performance of the OnDemand system.



Did you know

In this chapter, we offer various tidbits of information that was gathered in the course of writing this IBM Redbooks publication. Each section discusses a different feature of OnDemand that might be useful in administering a production environment.

We also provide some program samples, which you can download from the IBM Redbooks Web site. The samples offer practical ways to educate yourself on using OnDemand APIs or as base for more advanced development.

In this chapter, we cover the following diverse topics:

- ▶ Using the Document Audit Facility
- ▶ Related Documents feature
- ▶ Store OnDemand
- ▶ OnDemandToolbox
- ▶ Batch OnDemand commands in z/OS
- ▶ Testing PDF indexer in z/OS
- ▶ Date range search tip for users
- ▶ Ad-hoc CD-ROM mastering
- ▶ OnDemand Production Data Distribution
- ▶ Customizing the About window
- ▶ Modifying client behavior through the registry
- ▶ Negative numbers in decimal fields handling
- ▶ Message of the day
- ▶ OnDemand bulletins

15.1 Using the Document Audit Facility

OnDemand has incorporated a feature called the Document Audit Facility (DAF). This function allows for basic approval routing of a document. To use the DAF, you must first define the reports to be audited to OnDemand and create an audit control file. An administrator can define the default status for a document, and users with the appropriate permissions have the ability to click a button on the client to change the status of the document.

Background for our example

In our example, a company scans vendor invoices as they are received. Each invoice must be reviewed and approved before payment is made to the vendor.

An administrator sets up an index to the invoices that can be one of four values: Hold, Accept, Reject, or Escalate. When invoices are scanned, they are loaded with a default of Hold status. The only users who have permission to view these Hold invoices are the auditors or managers. After the auditor reviews the invoice, they can click a button to set the document to either Accept, Reject, or Escalate status:

- ▶ Accepted invoices should be paid.
- ▶ Rejected invoices should not be paid due to problems with the invoice.
- ▶ Escalated invoices should be reviewed by managers to determine if they should be paid.

This action changes the value of that index. Permissions for the Accounts Payable user group are set up in such a way that they can only view invoices that have the Accept status. Purchasing can view invoices with the Reject status to determine why they were rejected and contact the vendor to correct the problem. Auditors and managers can view invoices that have the Escalate status.

15.1.1 Creating a status field in the application group

Add a field to the application group that is called *audit* or *status*. This field must be a one-character string with Case set to Upper and the Type set to Fixed. In the Mapping section of the application group, add database values and displayed values for each of the required status. See Figure 15-1 for adding status field values to the application group. We added four values for status: Hold (H), Accept (A), Reject (R), or Escalate (E).

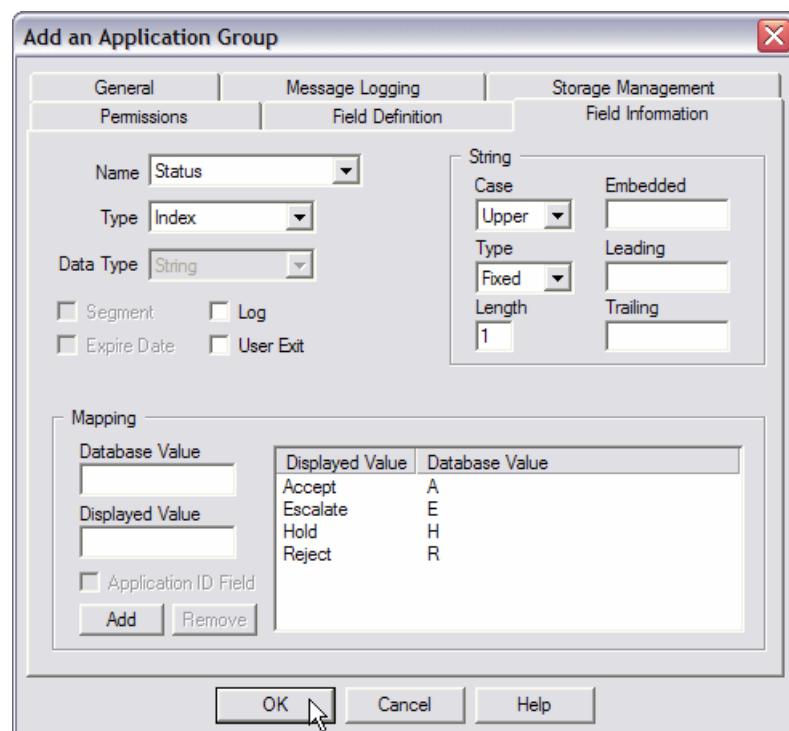


Figure 15-1 Adding status field to the application group

15.1.2 Setting a default in the application

Decide what you want the default value for this index to be and set this default in the application. For our example, invoices should be loaded to the system with Hold status, so we set the default to H. See Figure 15-2 for our example of setting the default status in the application on the Load Information tab.

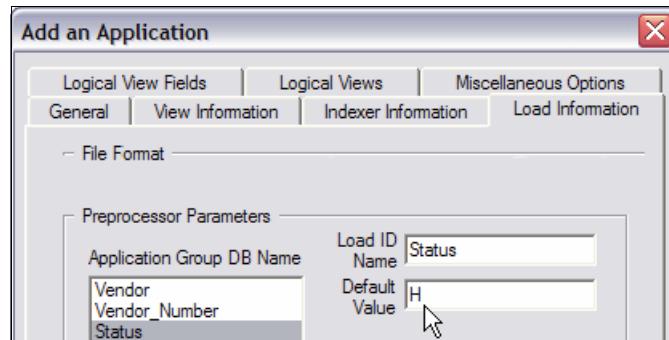


Figure 15-2 Setting default in application

15.1.3 Updating permissions for various users

Update the permissions for your users or user groups to restrict them to documents with specific status. In our example, Accounts Payable should only be allowed to view statements after they are audited or assigned an Accept (A) status. See Figure 15-3 for how to specify query restrictions.

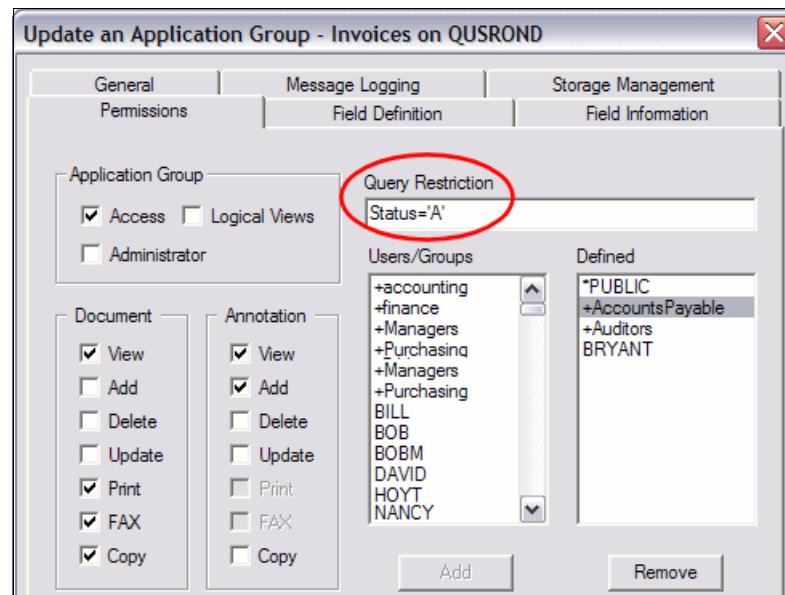


Figure 15-3 Query restriction

Query restrictions can be added at the user or user group level.

We also have a group of users named *Auditors*. These are people who are allowed to change the status of documents. This group has no query restriction and *must have update authority to the documents*, as shown in Figure 15-4.

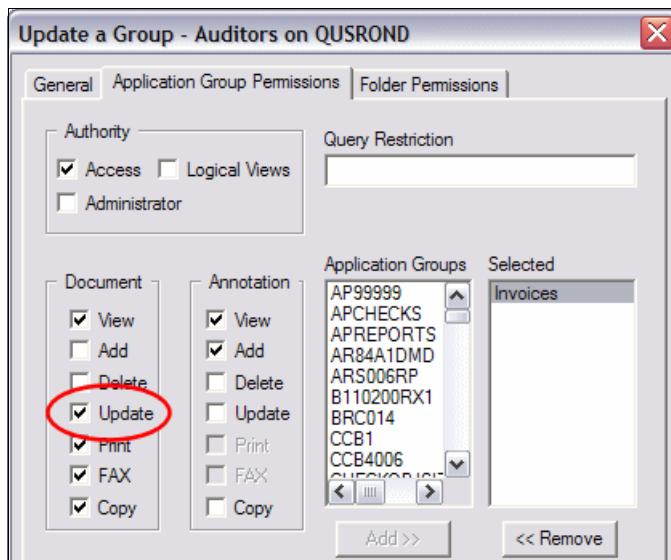


Figure 15-4 Application group permission: no query restriction

15.1.4 Creating folders for each user type

In our example, we created separate folders for the accounts payable view and the auditors view so that the status field is not visible from the accounts payable folder. The folder created for the auditor contained all the fields including the status of each document.

Note: To reduce administration, it might be advantageous to have a single folder, because it is possible to configure a folder to have different field views for different users.

15.1.5 Creating the Document Audit Facility control file

The Document Audit Facility is controlled by a file named ARSGUI.CFG, which you must create and store in the Windows client program directory (\Program Files\IBM\OnDemand32). The DAF file has the same format and syntax as a standard Windows INI file. This file contains a section named AUDIT, which identifies one or more folder sections. Each folder section identifies a folder that can be audited. See Figure 15-5 for our DAF control file.

```
[AUDIT]
Folders=Invoices

[Invoices]
FOLDER=Invoices - Auditor
AUDIT_FIELD=Status
TEXT1=Accept
TEXT2=Reject
TEXT3=Escalate
VALUE1=A
VALUE2=R
VALUE3=E
```

Figure 15-5 ARSGUI.CFG

AUDIT section

The AUDIT section contains one record, the FOLDERS record. The FOLDERS record contains a comma-separated list of folder section names. You must create an additional section in the DAF file for each folder section named in the FOLDERS record.

Important: The total number of characters in the FOLDERS record *must not exceed 255*.

FOLDER section

Each FOLDER section contains the following records:

- ▶ FOLDER specifies the name of the folder, exactly as it appears in OnDemand. The FOLDER record is required.
- ▶ AUDIT_FIELD specifies the name of the folder field used to audit documents, exactly as it appears in OnDemand. The AUDIT_FIELD record is required.

- ▶ TEXTx is the caption that appears on the command button used to change the status of the document. Up to eight TEXT settings are permitted.
- ▶ VALUEx is the value that is stored in the database when the corresponding TEXTx button is clicked. This value is stored in the application group field and must match one of the mapped field values. One VALUE record is required for each TEXT record. Up to eight VALUE settings are permitted.

Note: You must restart the OnDemand client after you create this file.

15.1.6 Viewing the folders

Now when the auditor logs into the Invoices - Auditor folder, three buttons (on the bottom right) allow for updating of the status field. See Figure 15-6.

Statement Date	Account	Account Name	Balance	Status
03/03/2002	000-000-006	SPORTS CO	2828.11	Pass
04/03/2002	000-000-006	SPORTS CO	2020.11	Pass
07/03/2002	UUU-UUU-UU6	SPORTS CO	2311.95	Pass
08/03/2002	000 000 006	SPORTS CO	1651.77	Fail
06/03/2002	UUU-UUU-UU6	SPORTS CO	2311.95	Hold
09/03/2002	000-000-006	SPORTS CO	43.62	Fail
10/03/2002	000-000-006	SPORTS CO	2003.75	Hold
11/03/2002	000-000-006	SPORTS CO	2003.75	Pass
01/03/2002	000-000-006	SPORTS CO	2605.00	Pass
02/03/2002	000-000-006	SPORTS CO	2605.33	Hold

Figure 15-6 Auditors view of the folder

When Accounts Payable users query the server, they are limited in what they can see. They do not see the status buttons or the status of the documents. Accounts Payable only sees the statements that are accepted by the auditor (Figure 15-7).

Document List				
	Invoice Date	Vendor	PO #	Invoice #
	11/18/2005	HARDWARE SHOP	A13456	983356
	11/15/2005	COMPUTER SHOP	A14211	0511-11
	11/11/2005	CONCRETE SHOP	A14533	5422
	11/28/2005	TACK SHOP	A14222	004888
	11/28/2005	CAR INC	A14245	11J1
	12/05/2005	LUMBER INC	A12545	68555
	12/07/2005	PRODUCTS INC	A12345	1000-IO

Append AutoScroll

Figure 15-7 Customer view of the folder

This document audit facility might be a useful way of auditing documents in OnDemand. We only set up four status field values in our example. There is a *limit of eight status buttons* per folder.

To take this example further, it is possible to set up multiple folders each with their own distinct status buttons. By doing so, it is possible to route a document through a series of auditing. You can define various users, each with a different auditing responsibility.

For example, a particular user is responsible for pulling up all failed documents and placing them in some other status. To do this, each status must be mapped in the application group, and each folder must be specified within the appropriate user's arsgui.cfg file. It is also important to define each user with the correct search and update permissions in the application group.

15.1.7 Tracking changes to invoice status

You might want to track who is changing the status of the invoices. This is often a requirement of financial auditors in a company. This is done in the Application Group by selecting to log document updates and to log specific field values.

On the Application Group, the Message Logging tab, verify that **Index Update** is selected. Selecting this option causes system log message 80 to be logged every time an invoice index is updated. See Figure 15-8.

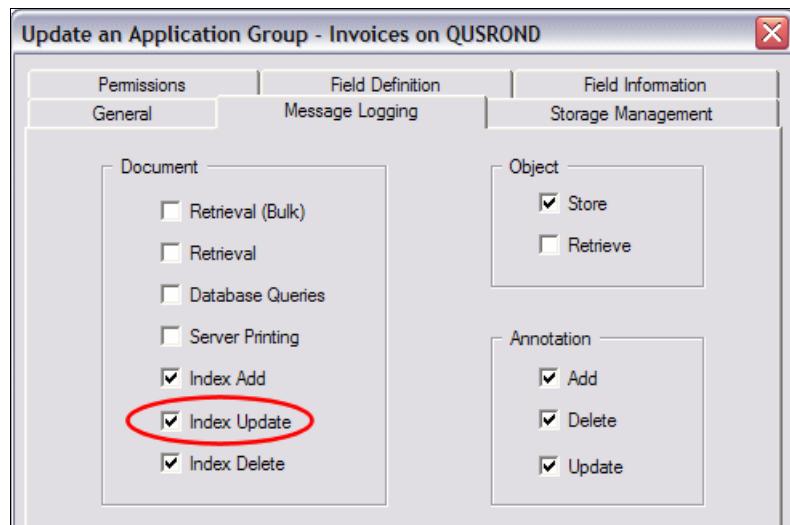


Figure 15-8 Message logging setup

In the Update an Application Group window, on the Field Information tab, select the **Log** check box for each field whose value you want captured when the invoice status is updated. See Figure 15-9.

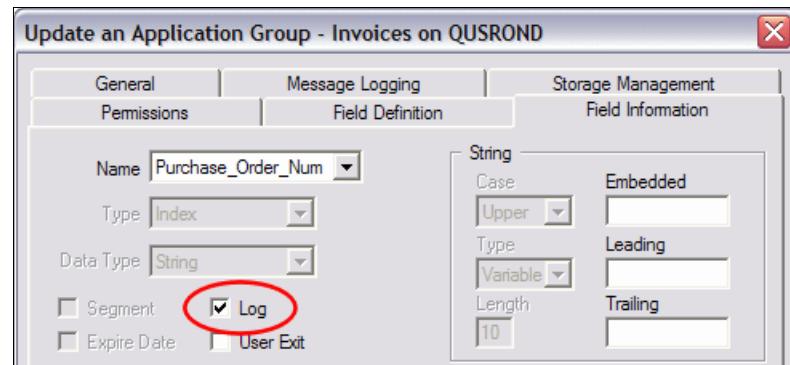


Figure 15-9 Field update logging setup

In our example, we log three field values so that we can uniquely identify the invoice as accepted or rejected. We log the purchase order number, invoice number, and status.

To check who is updating the invoice status, use the system log to review message number 80, which contains the information about Application Group document updates. System log message 80 includes the date and time the update was made, the user ID making the update, and message text of the update.

The following example shows a complete system log message 80:

```
01/09/2006 00:18:02DUSER1      40376InfoN/A      80ApplGroup DocUpdate:  
Name(Invoice) Agid(5395) OrigFlds() UpdFlds()
```

If no fields are selected for logging, the system log 80 message contains blanks, as shown in the following example:

```
ApplGroup DocUpdate: Name(Invoice) Agid(5395) OrigFlds() UpdFlds()
```

If only the status field is selected for logging, the system log 80 message contains only the before and after status values. This is not sufficient information to identify the exact document rejected, as shown in the following example:

```
ApplGroup DocUpdate: Name(Invoice) Agid(5395) OrigFlds('H') UpdFlds('R')
```

In our example the purchase order number, invoice number, and status field are selected for logging. The system log 80 message contains enough information to identify the exact invoice rejected, as shown in the following example:

```
ApplGroup DocUpdate: Name(Invoice) Agid(5395) OrigFlds('A12611','762301','H')  
UpdFlds('A12611','762301','R')
```

15.2 Related Documents feature

The *Related Documents* feature is a relatively unknown but useful feature that is available within the standard OnDemand Windows client. This feature give users the ability to retrieve a document and then, based on the content of that document, they can search for other related documents stored in OnDemand with a simple click of a button.

For details about how to configure the Windows client for Related Documents, refer to the “Related Documents” section, in the “Windows 32-bit GUI Customization Guide” chapter, in *IBM Content Manager OnDemand - Windows Client Customization Guide and Reference*, SC27-0837.

The following sections contain extracts from this guide along with important tips and practical examples of how to configure this feature.

15.2.1 How it works

In principle, the Related Documents feature is designed so that you can load two completely different types of documents in OnDemand and link them together within the OnDemand client. For example, a hypothetical finance company produces quarterly finance reports for each of its clients. In conjunction with these reports, the company produces a summary sheet containing a subset of reference numbers for these financial reports. It is possible using the Related Documents feature to access the financial reports with a single click from within the summary sheet.

The way in which this is done is by the user selecting text from within a document and then clicking the Related Documents icon, which is on the task bar when Related Documents is configured for that folder. The text that is selected is then used as the search criteria on another folder. The first document in the hit list from this search is displayed in the client along side the document already open.

Using the preceding example, the *summary sheet* must be a document type that allows text to be selected within the OnDemand client; Related Documents does not work if the *summary sheet* is either PDF or image data.

Figure 15-10 shows an OnDemand Windows client that is configured to use the Related Documents feature and illustrates a line data document that has been used to retrieve a letter and a credit card statement using the Related Documents icon. The Related Document icon for this example is a star on the right side of the toolbar. You can replace this icon with any icon design that you choose.

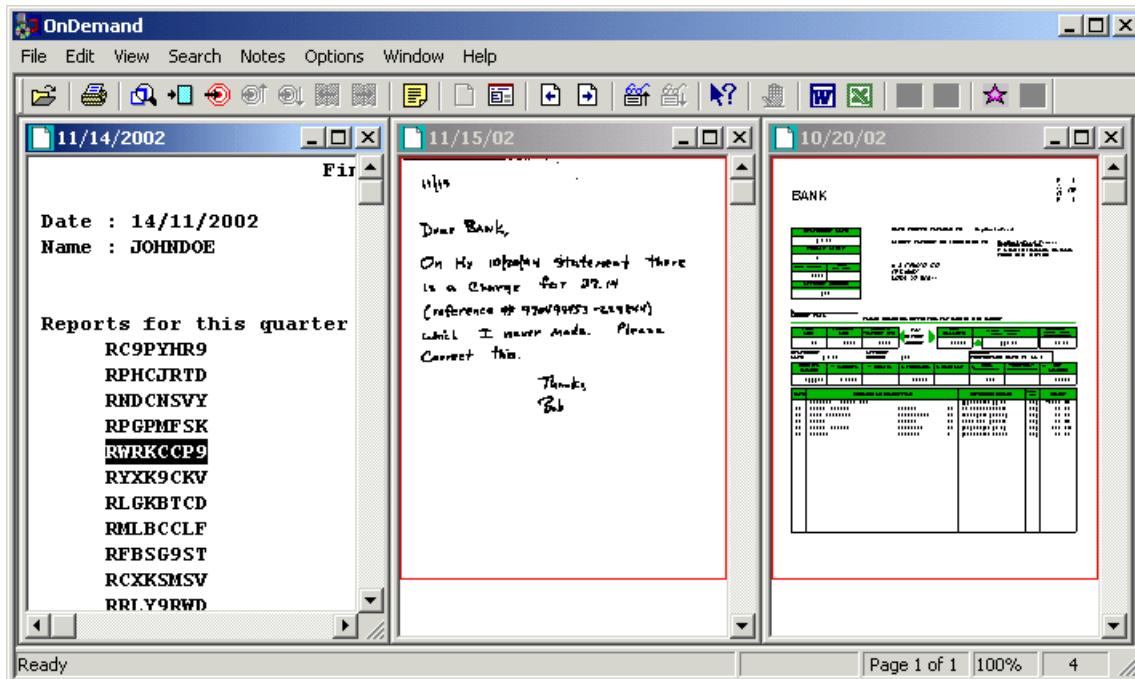


Figure 15-10 Related Documents search

15.2.2 Configuring Related Documents

To configure an OnDemand Windows client to enable the Related Documents feature, it is necessary to edit the registry of the Windows machine on which the client is installed. For a detailed description of the registry keys and string values that must be added, see *IBM Content Manager OnDemand - Windows Client Customization Guide and Reference*, SC27-0837.

We provide a sample registry file (import.reg) in Example 15-1, which can be edited with your own folder names, fields, and icon values and then imported into the registry.

Example 15-1 Sample registry import file (import.reg)

Windows Registry Editor Version 5.00

```
[HKEY_CURRENT_USER\Software\IBM\OnDemand32\Client\RelatedDocs]
"Related"="Letters,Bank"

[HKEY_CURRENT_USER\Software\IBM\OnDemand32\Client\RelatedDocs\Bank]
"MenuText"="Related Check"
"BitmapDLL"="c:\\reldocs\\extadd11.dll"
"BitmapResid"="135"
"RelatedFolder"="Cheque Images"
"Fields"="Amount=eq\\%s"
"Arrange"="v"
"Folders"="Bank*\\Credit*"

[HKEY_CURRENT_USER\Software\IBM\OnDemand32\Client\RelatedDocs\Letters]
"MenuText"="Financial Report"
"BitmapDLL"="c:\\reldocs\\extadd11.dll"
"BitmapResid"="135"
"Folders"="Letters"
"RelatedFolder"="Financial Report"
"Fields"="Reference Number=eq\\%s"
"Arrange"="v"
```

To import a file into the registry:

1. Create a registry file using the sample in Example 15-1. The file should have the extension of .reg.
2. Click **Start** → **Run....**
3. Type regedit into the box that is displayed.
4. In the register editor tool, click **Registry** → **Import Registry File**.
5. Select the registry file that you created in step 1 and click **Open**.

After you import the registry keys and string values, the structure of this part of the registry should look similar to the example in Figure 15-11.

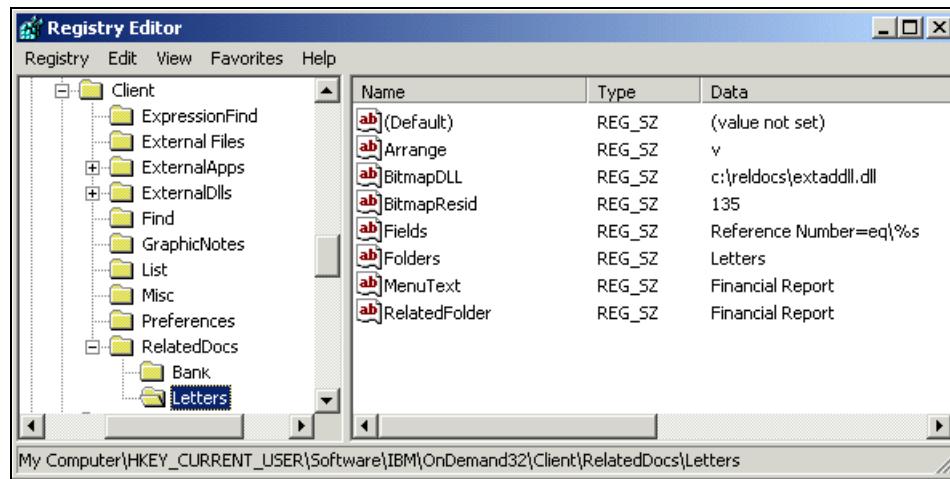


Figure 15-11 Registry structure for Related Documents feature

15.3 Store OnDemand

Store OnDemand is a graphical application that allows a user to index and store any PC document into an existing OnDemand server. It is designed for the storing of single documents rather than the main design point of OnDemand, which is to load large quantities of report files in batch. With the multipurpose store capabilities of Store OnDemand, a user can quickly and easily archive any document that is then instantly available to the rest of the enterprise.

Store OnDemand stores documents to OnDemand servers on UNIX, Windows, z/OS, and iSeries servers. It operates as a thick client to access OnDemand (V7.1) and runs on the Microsoft Windows platforms. It requires a DB2 Universal Database client to be installed on the PC and a database-to-database connection to the server.

Store OnDemand is available as sample with the source code. It is intended to provide customers and partners examples of how to code APIs while providing a useful as-is tool to meet customer needs.

Note: We provide the Store OnDemand executable and the source code as an example to show you what you can implement with the APIs. They are strictly to be used “as is.” The executables and the source code are *not* supported by the IBM United Kingdom developers who developed the code, the IBM DB2 Content Manager support team, or the IBM Redbooks team. Refrain from contacting the developers and the support personnel about these items.

To download the executables and the source code, refer to Appendix A, “Additional material” on page 605.

15.3.1 Why it is needed

Previously, to load a single document into OnDemand, it was necessary to load it via the Generic Indexer. The basic process to follow is to produce an index file similar to Example 15-2. Then with the index file and the document file, the **arsload** program is used to load the document into OnDemand.

Typically, the only place that the **arsload** program can run is on the server. Therefore users might send their files that require loading into OnDemand to an administrator who initiates a daily batch load of these documents. With the Store OnDemand services offering, you no longer need to rely on a system administrator to store a single document. The following section explains what the service offering provides.

Example 15-2 Sample Generic Index file

```
COMMENT: OnDemand Generic Index File Format
COMMENT: This file has been generated by the arsdoc command
COMMENT: February 25, 2002 09:04:04
COMMENT:
CODEPAGE:5348
COMMENT:
IGNORE_PREPROCESSING:1
COMMENT:
GROUP_FIELD_NAME:name
GROUP_FIELD_VALUE:BANK
GROUP_FIELD_NAME:account
GROUP_FIELD_VALUE:000-000-000
GROUP_FIELD_NAME:crd_date
GROUP_FIELD_VALUE:19950303
GROUP_FIELD_NAME:balance
GROUP_FIELD_VALUE:1058.110000
GROUP_OFFSET:0
GROUP_LENGTH:2800
GROUP_FILENAME:c:\temp\credit.1.BANK Credit.BANK Credit.out
```

15.3.2 What it does

Store OnDemand is a client-based application that allows users to initiate the storing of documents into OnDemand directly rather than going through an administrator. To store a document in OnDemand:

1. Launch the Store OnDemand application. The user is required to logon using an OnDemand user ID and password.
2. In the Store OnDemand window (Figure 15-12) that opens, complete the following steps:
 - a. Locate the document that the user want to store into OnDemand. Either click **Locate** to find the document, or look in a Windows Explorer window and drag it onto the Store OnDemand window.

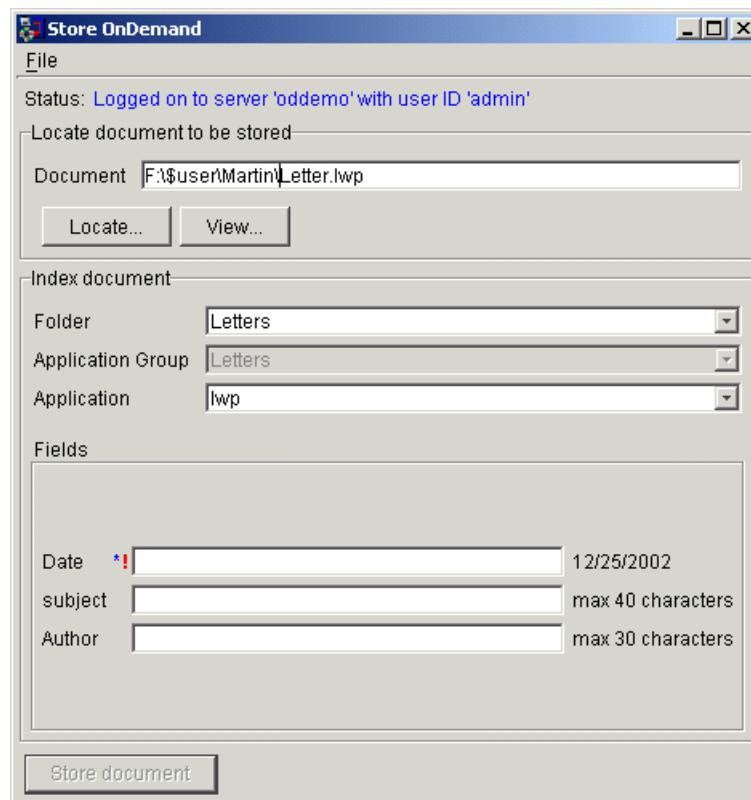


Figure 15-12 Store OnDemand screen display

- b. From the Application group and Application lists, select the application group and application in which the document should be stored.

- c. When the application group is selected, the index fields required for this document are displayed. Either enter the index fields or click **View** and cut and paste the index values from the document to reduce the opportunity for user error.
- d. Click **Store Document** to load the document into OnDemand.

Note: Data verification functions within Store OnDemand ensure that data entered is in the correct format and type to further reduce the possibility of entering incorrect indexes. The Store Document bottom is unavailable until this validation criteria is satisfied. Otherwise, storing is not possible. Guidance regarding this criteria is provided in the Store OnDemand window.

15.4 OnDemandToolbox

IBM Germany developed a set of sample programs for use with OnDemand Common Server. These programs are intended to provide customers and partners examples of how to code client APIs while providing a useful as-is tool to meet customer needs. The sample toolbox code and documentation are available on the IBM Redbooks Web site. Refer to Appendix A, “Additional material” on page 605, for download instructions.

The components of the toolbox are:

- ▶ OD Update
- ▶ OD Delete
- ▶ OD Store
- ▶ OD File System Monitor

Each component is a single program, working as stand-alone application. All programs are written in Visual Basic®.net and are provided as ready-to-use binaries and as source code under the IBM Public License. Installation of the OnDemand client is a prerequisite for the installation of the OnDemand Toolbox. The Microsoft .NET Framework must be installed during the setup of the toolbox.

OD Update

The OD Update application provides an easy-to-use interface to update the index (key) values of documents archived in OnDemand.

The key features are:

- ▶ Easy update of all documents
- ▶ No need for any client configuration before using it
- ▶ Fits into the OnDemand security concept

The user must have update authority to the application group, which is configurable using the OnDemand Administrator.

OD Delete

The OD Delete application is used for deleting documents archived in OnDemand.

The key features are:

- ▶ Easy deletion of single or multiple documents
- ▶ No need for any client configuration before using it
- ▶ Fits into the OnDemand security concept

The user must have delete authority to the application group, which is configurable using the OnDemand Administrator.

Both, OD Update and OD Delete applications provide a user interface that is similar to the OnDemand client, making them easy to use.

OD Store

The OD Store application is used for archiving PC files directly into OnDemand.

The key features are:

- ▶ Easy interface for archiving single documents into OnDemand
- ▶ Integrates with the Windows Explorer
- ▶ Can handle all file types supported by OnDemand
- ▶ Requires prior configuration of which file type should be archived using which application, application group, and folder
- ▶ Configuration can be exported and easily distributed to other users.
- ▶ Fits into the OnDemand security concept

The user must have archive authority to the application group, which is configurable using the OnDemand Administrator.

OD File System Monitor

The OD File System Monitor application is a background application that monitors directories on a PC and automatically archives the files found in those directories using preconfigured field information and information extracted from the files.

To use the monitor, configure the following items:

- ▶ The directories that should be monitored (multiple directories and network drives supported)
- ▶ The file types that should be archived into OnDemand
- ▶ What to do after successful or unsuccessful archiving
- ▶ The file types that should be archived into which application and which application group
- ▶ What to do with files of other, unknown file types found in the monitored directories
- ▶ The folder that should be used and the information that should be used for the indexes

The key features are:

- ▶ When configured, the OD File System Monitor runs without any user interaction as a background process.
- ▶ The document's values for each index can be set to the file's meta information such as file size, creation date and time, and more.
- ▶ The OD File System Monitor creates detailed logs (configurable) about all its activities.
- ▶ The OD File System Monitor fits into the OnDemand security concept.
The configured user account must have archive authority to the application group, which is configurable using the OnDemand Administrator.
- ▶ Configuration of the OD File System Monitor is saved as an XML file, and therefore, can be distributed among other installations.
- ▶ The source code can easily be modified to extract index information from other sources such as the file's content.

15.5 Batch OnDemand commands in z/OS

The OnDemand commands are designed to be entered from a UNIX or Windows NT command line. The ARSLOAD and ARSADMIN commands can be called as a program with the proper job control language (JCL) provided and the output is written to DD sysout. Other commands, such as ARSDATE and ARSADM, can be run as a batch job using the BPXBATCH program.

The example in Figure 15-13 shows how to run the ARSDATE command as a batch job. The executable file ARSDATE resides in the /usr/lpp/ars/bin directory in the hierarchical file system (HFS) of the UNIX System Services. The output is written to the STDOUT statement, which points to an outfile in the HFS directory /tmp/arssockt.std. This file must be accessible or the user running the BPXBATCH program must have proper authority to create this file.

```
//ARSDATE JOB (QFTA0000,B123),  
//           'JOHN SMITH',MSGCLASS=0,CLASS=U,  
//           NOTIFY=&SYSUID,USER=TEAM5  
//STEP1      EXEC PGM=BPXBATCH,REGION=0M,  
// PARM='PGM /usr/lpp/ars/bin/arsdate 11/23/02'  
//STEPLIB    DD DISP=SHR,DSN=OD390.V710.DBS.SARSLOAD  
//           DD DISP=SHR,DSN=ICCDDB2.SDSNEXIT  
//           DD DISP=SHR,DSN=ICCDDB2.SDSNLOAD  
//SYSPRINT   DD SYSOUT=*<br/>  
//SYSABEND   DD SYSOUT=*<br/>  
//SYSOUT     DD SYSOUT=*<br/>  
//DSNAOINI   DD PATH='/etc/ars/cli.ini'  
//STDERR     DD PATH='/tmp/arssockt.stderr',PATHMODE=SIRWXU,  
//           PATHOPTS=(OWRONLY,OCREAT,OAPPEND),PATHDISP=(KEEP,KEEP)  
//STDOUT     DD PATH='/tmp/arssockt.stdout',PATHMODE=SIRWXU,  
//           PATHOPTS=(OWRONLY,OCREAT,OAPPEND),PATHDISP=(KEEP,KEEP)  
//STDENV     DD PATH='/usr/lpp/ars/config/arssockd.stdenv',  
//           PATHOPTS=ORDONLY
```

Figure 15-13 BPXBATCH sample program

15.6 Testing PDF indexer in z/OS

A good way to test the PDF indexer is to run the indexer as a batch job without loading the data to the database with the ARSPDOCI program. The ARSPDOCI is shipped with the basic code. Your instance ARSSOCKD must not be up. The JCL sample in Figure 15-14 shows how to run the ARSPDOCI program.

```
//ARSPDFLO JOB (QFTA0000,B123),
//      'JOHN SMITH',MSGCLASS=0,CLASS=U,
//      NOTIFY=&SYSUID,USER=TEAMS
//STEP1   EXEC PGM=ARSPDOCI,REGION=0M,
// PARM='parmdd:DDN:PAR OUTPUTDD=hfs:/tmp/pdf.out INDEXDD=hfs:/tmp/pdf
//          .ind'
//STEPLIB   DD DISP=SHR,DSN=TEAM5.ODMP710.SARSLOAD
//SYSPRINT DD SYSOUT=*
//***** PDF indexer parmfile contains the indexer parameters *
//***** PDF indexer files.
//ADOBERES  DD DSN=TEAM5.PDFLIB.RESOURCE.INDEX(ADOBERES),DISP=SHR
//ADOBEPNT  DD DSN=TEAM5.PDF405.PLUSP1C.ADOBEPNT.LST,DISP=SHR
//TEMPATTR  DD DSN=TEAM5.PDF405.PLUSP1C.TEMPATTR,DISP=SHR
```

Figure 15-14 ARSPDOCI sample JCL

The PAR DD file contains the parameter for the indexer (Example 15-3). You can cut and paste this information from the Application panel indexer information.

Example 15-3 Parameter file for ARSPDOCI program

```
COORDINATES=IN
TRIGGER1=UL(5.67,0.85),LR(6.18,1.09),*, 'Page 1'
FIELD1=UL(4.68,0.85),LR(5.71,1.06),0,(TRIGGER=1,BASE=0)
FIELD2=UL(4.17,1.22),LR(7.39,1.47),1,(TRIGGER=1,BASE=0)
FIELD3=UL(5.61,1.43),LR(6.49,1.71),1,(TRIGGER=1,BASE=0)
FIELD4=UL(5.14,1.46),LR(5.61,1.64),1,(TRIGGER=1,BASE=0)
INDEX1='rdate',FIELD1,(TYPE=GROUP)
INDEX2='rname',FIELD2,(TYPE=GROUP)
INDEX3='rplan',FIELD3,(TYPE=GROUP)
INDEX4='rrefcode',FIELD4,(TYPE=GROUP)
INDEXSTARTBY=10
```

INDEXDD=hfs:/tmp/pdf.ind
INPUTDD=DDN:INPUT

15.7 Date range search tip for users

Instead of keying specific dates when searching for documents within the OnDemand client, it is sometimes easier to use the *T date search option*. This option lets you search for documents based on the days, months, or years that are relative to the current system date of the PC.

Type the letter T in the date search field and set the search operator to Equal To. When you run the search, OnDemand retrieves the documents that contain today's date.

The T date search option might be used with the search operator set to Between or set to Equal To. You can also use the following patterns when you use the T date search option:

T { + or - } # { D or M or Y }

The braces denote groups of optional parameters for the T format string; choose one of the symbols in the group. If you leave out the plus sign (+) or the minus sign (-), OnDemand assumes a + sign. If you leave out D, M, or Y, OnDemand assumes D. The T format string is case insensitive, meaning that you can type T or t, D or d, M or m, or Y or y.

Table 15-1 describes the T format string.

Table 15-1 T format string

Symbol	Description
T	Current date (required)
+	Forward from the current date
-	Backward from the current date
#	Represents the number of days, months, or years (required)
D	Days (number of days)
M	Months (number of months)
Y	Year (number of years)

Table 15-2 lists examples of using the T format string with the search operator set to Equal To.

Table 15-2 T format string examples with the Equal To search operator

T string	Meaning
T-6M	6 months prior to the current date
T+30D	30 days forward from the current date
T30	30 days forward from current date (same as above)
T-1Y	1 year prior to the current date

Table 15-3 lists examples of using the T format string with the search operator set to Between.

Table 15-3 T format string examples with the Between search operator

T string		T string	Meaning
T-6M	and	T	Between 6 months prior to the current date and the current date
T-60D	and	T-30D	Between 30 and 60 days prior to the current date
T-60	and	T-30	Same as previous
T	and	T30	Between the current date and 30 days forward from the current date

15.8 Ad-hoc CD-ROM mastering

You can extract data from an OnDemand server and use the OnDemand client to put that data onto a directory on your PC. You can then copy the documents and indexes to a CD or DVD for easy distribution, or leave the data in your PC directory for demonstration purposes.

Select **Ad-Hoc CDROM Mastering** during the client installation (see Figure 15-15). Then follow the steps in the rest of this section.



Figure 15-15 Client installation: selecting the Ad-Hoc CDROM Mastering option

15.8.1 Transferring documents from the OnDemand server to the staging drive

Follow these steps to use the CD-ROM mastering option:

1. Launch the OnDemand client, and select **File → Set CD-ROM Mastering Options**. See Figure 15-16.

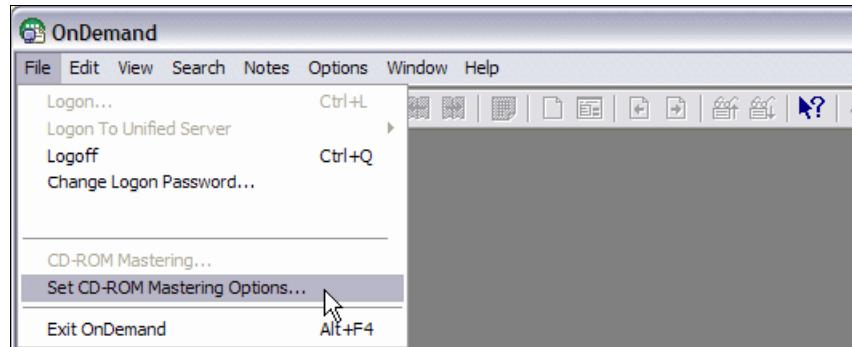


Figure 15-16 Set CD-ROM Mastering Options

2. In the Set CD-ROM Mastering Options window (Figure 15-17), under CD-ROM User, enter your user ID and password to the CD-ROM, which is an OnDemand server. The default user ID and password are both cdrom. You must select the staging path. Then click **OK**.

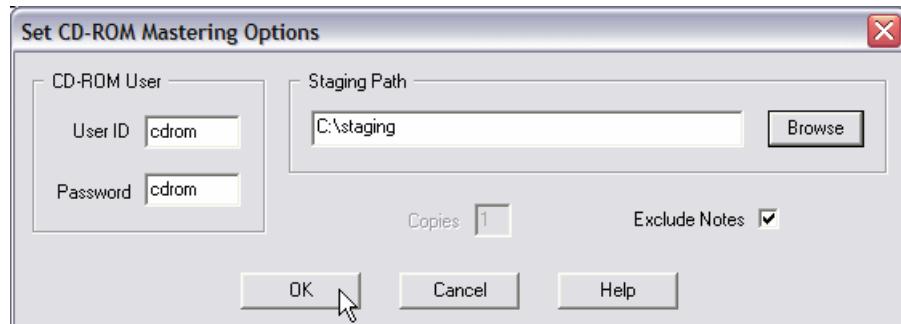


Figure 15-17 Set CD-ROM Mastering Options: staging path

Important:

- ▶ If the Exclude Notes check box is checked, public annotations are *not* copied.
- ▶ Only local hard drives are available as staging drives.
- ▶ You cannot select more than one copy.

3. The CD-ROM Mastering option becomes active on the File menu.
4. Run a search and the document list is displayed. Then select **File** → **CD-ROM Mastering**. See Figure 15-18.

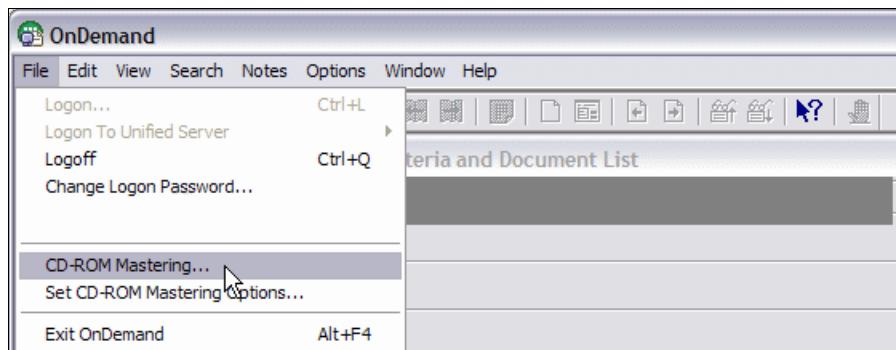


Figure 15-18 Starting the CD-ROM Mastering feature

Important: The CD-ROM Mastering option is available only when a document list is displayed. If a document in the list is being viewed, the CD-ROM Mastering option is unavailable.

5. In the CD-ROM Mastering window (Figure 15-19), follow these steps:
 - a. Select the folder that you want to add to CD-ROM. Your current folder is automatically selected, and you can select alternate folders from the list.
 - b. In the CD-ROM folder description field, enter a description. The description is required and is saved in the drop-down list for the user in other mastering sessions.
 - c. You can update the CD-ROM mastering options by choosing the **Set Options** button.
 - d. Click **OK**.

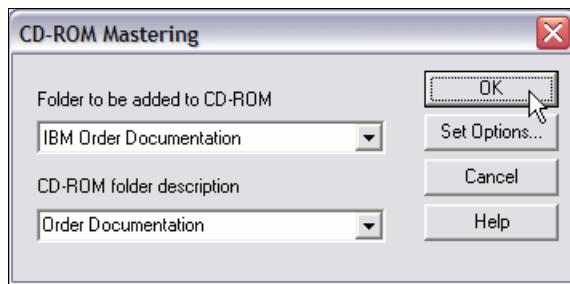


Figure 15-19 CD-ROM Mastering: adding folders to CD-ROM

Important:

- ▶ To transfer the documents to a staging drive, you must keep the document list on the screen. Only one folder can be staged at a time.
- ▶ All items in the document list are placed on the CD-ROM.

6. The CD-ROM mastering process starts. You should be able to see a window with five options in it (Figure 15-20):
- **Clean:** Removes all files in the staging directory
 - **Setup:** Creates the necessary directory structures in the staging directory
 - **Fetch:** Retrieves the data and resources for the items in the hit list
 - **Index:** Re-indexes the retrieved data for the CD-ROM
 - **Stage:** Copies the CD-ROM installation files and the OnDemand client (along with any installed languages) to the staging directory

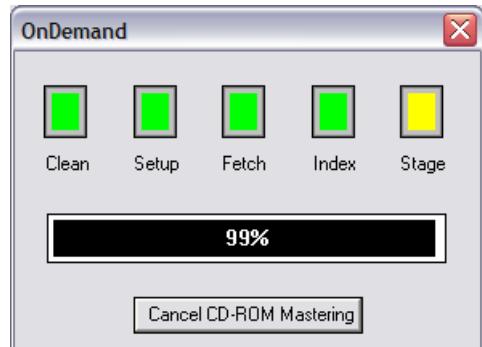


Figure 15-20 CD-ROM Mastering process

7. After the CD-ROM mastering process finishes, you see a message like the example in Figure 15-21. Click **Yes** to finish the process. Click **No** if you want to add another folder or stop the CD-ROM mastering process.

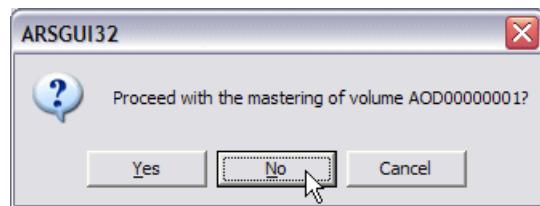


Figure 15-21 CD-ROM Mastering: process complete message

8. If you click **No** in the previous step, you see the message shown in Figure 15-22. Click **No** to finish the process. Click **Yes** to return to the previous window to select another folder.

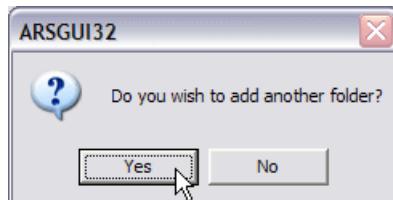


Figure 15-22 CD-ROM Mastering: adding another folder

9. After you finish selecting the folders and staging documents you want, when prompted by the message "Proceed with the mastering of volume xxxxxxxxxxx?", click **Yes**. The CD-ROM image is finalized and can now be accessed with the OnDemand client or written to a CD-ROM.

OnDemand writes messages into the system log while documents are retrieved for the CD-ROM image. After the CD-ROM image is finalized, a CD-ROM creation manifest is written that contains the user ID, password, and a listing of the folders that are included in the CD-ROM image.

The following examples are system log messages created by CD-ROM mastering:

- ▶ **Message 81**

```
ApplGroup ObjRetrieve: Name(CHECKS) Agid(5252) ObjName(4FAAA)
NodeName(-CACHE-) Nid(4) Server(-LOCAL-) Off(0) Len(68928) Time(0.001)
```

- ▶ **Message 67 (AFP applications only)**

```
ApplGroup ResGet: Name(ABINVRX1) Agid(5027) NodeName(-CACHE-) Nid(0)
Server(-LOCAL-) Time(0.066)
```

- ▶ **Message 90**

```
BULK DOCUMENT RETRIEVAL
```

```
Application Group Agid Flds->Handle
```

```
-----  
CHECKS      5252  ->4FAAA,0,8154,0,68928,0x4E,0x4F,0,4,0  
              ->4FAAA,8154,16464,0,68928,0x4E,0x4F,0,4,0  
              ->4FAAA,24618,12483,0,68928,0x4E,0x4F,0,4,0  
              ->4FAAA,37101,9862,0,68928,0x4E,0x4F,0,4,0  
              ->4FAAA,46963,4669,0,68928,0x4E,0x4F,0,4,0  
              ->4FAAA,51632,3590,0,68928,0x4E,0x4F,0,4,0
```

► Message 89 (CD-ROM Creation Manifest)

CD-ROM Volume A0D00000002

Produced on Friday, December 16, 2005 at 10:52:18 Eastern Standard Time by
USER1

COPIES 1

USER cdrom
PASSWORD cdrom

FOLDERS IBM Order Documentation
Generic Indexer Images
Invoice
CHECKS
ABINVRX1
AFPLO
JOBLOGLO

15.8.2 Burning the disk image to the optical media

Use the CD-ROM authoring software of your choice to burn the staging drive to a CD or DVD. The staging drive is as specified in the arsmstr32.ini file. Popular CD-ROM authoring software includes Roxio Ez-CD Creator, Nero, and Stomp.

15.8.3 Accessing the CD image from disk

To access the CD-ROM image from the local disk drive:

1. Open the OnDemand client.
2. Click **Update Servers**.
3. In the Update Servers window (Figure 15-23), add a local server.

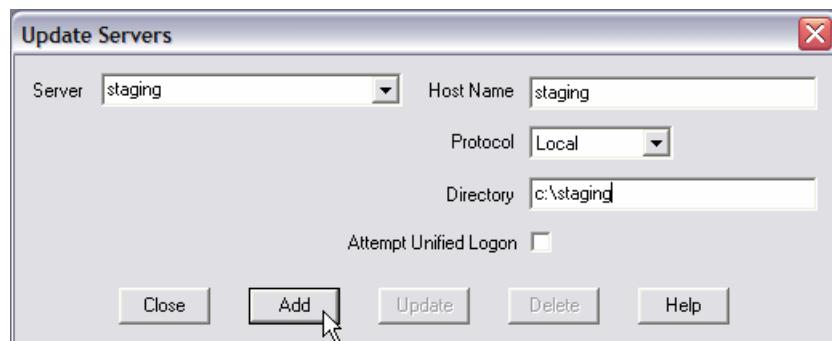


Figure 15-23 Accessing a CD image from disk: updating servers

4. Log on with the user ID and password specified when the CD-ROM image was created (Figure 15-24).

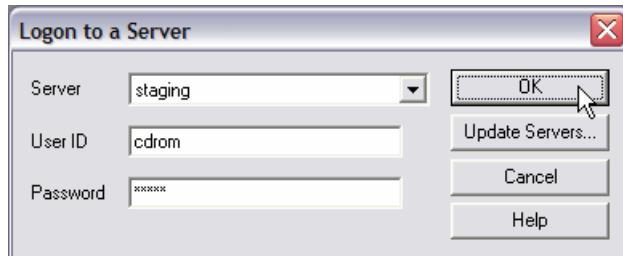


Figure 15-24 Accessing CD image from disk: log in

5. Use the OnDemand client to access the data in the usual way (Figure 15-25).

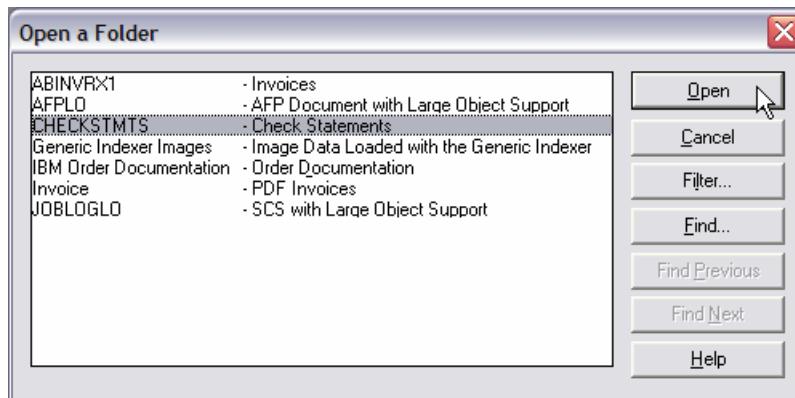


Figure 15-25 Accessing CD image from disk: opening a folder

15.8.4 Accessing the CD image from the CD-ROM

To access the CD-ROM image from the CD-ROM drive:

1. Run the setup program from the CD: x:\client\windows\win32.
2. Follow the prompts, which are similar to the regular OnDemand client installation. All languages from the original OnDemand client installation are available.

IBM OnDemand CDROM has been added to the Windows Start → All Programs menu. To start it, select **Start → All Programs → IBM OnDemand32 CDROM → OnDemand32 English**.

3. Log on with the user ID and password specified when the CD-ROM image was created (Figure 15-26).

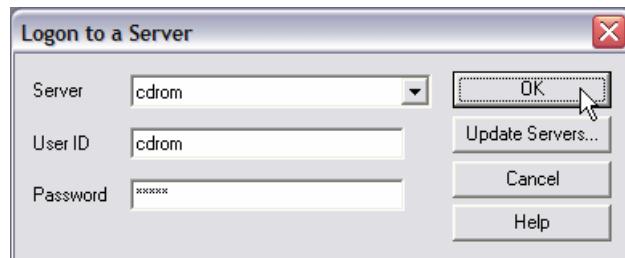


Figure 15-26 Accessing CD image from CD-ROM: logging in

4. Open a folder and use the OnDemand client to access the data in the usual way (Figure 15-27).

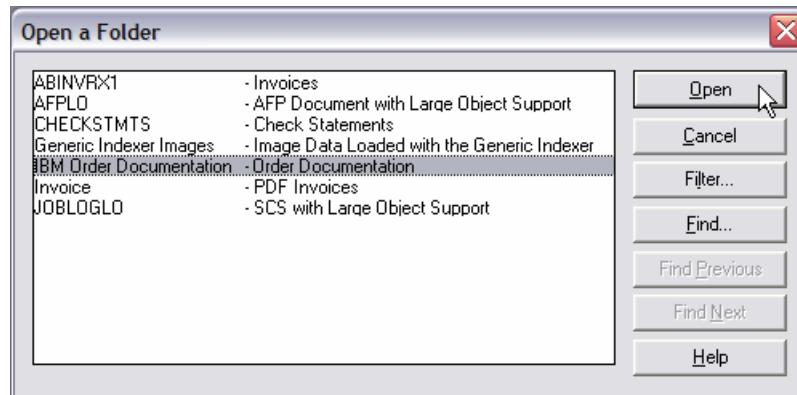


Figure 15-27 Accessing CD image from CD-ROM: opening a folder

A few differences are noted here:

- The logical AND/OR radio buttons are not displayed.
- The password cannot be changed.
- The annotations and named queries are stored on the user's disk drive, not on the CD-ROM.

15.9 OnDemand Production Data Distribution

The *OnDemand Production Data Distribution* (PDD) feature is an optional feature of OnDemand that you can use to distribute information to customers and other people inside and outside your company. To access the information, users mount the CD-ROM that has the OnDemand data and start an OnDemand client program that is included on the CD-ROM. Authorized users can view, reprint, or FAX documents that you distribute on the CD ROM.

15.9.1 Why it is needed

The PDD feature is designed to support high-volume, batch processing of input files and documents and production of multiple copies of a distribution image. The usage of the PDD CD-ROM is similar to the ad-hoc CD-ROM feature. The PDD feature complements the existing OnDemand recordable CD-ROM feature, which is designed for low-volume, ad-hoc building of CD-ROMs initiated by a user with OnDemand client programs. The PDD process is initiated from the OnDemand server instead of the client.

15.9.2 How it works

The PDD uses the Perfect Image Producer system (Rimage System) by Rimage Corporation to produce an ISO-9660 format disc image and to control the CD recorders, disc transporter, and CD label printer. The Rimage system includes the recorders, transporter and printer in one enclosure. It is also installed with the premastering, recording, and printing software.

The content of a distribution image is determined by the system administrator via distribution files and distribution groups. The system administrator can design the label of the CD-ROM as something meaningful because the CD-ROM image produced by the PDD program can be printed with customized color label using the Rimage System.

PDD is a services offering. For more information about the Production Data Distribution, contact your local IBM marketing representative.

15.10 Customizing the About window

The About window is displayed when the OnDemand client is first started and can be displayed later by selecting **Help → About OnDemand**. The About window can be customized with user-specific text and graphics. Up to eight lines of text can be added as well as a bitmap and text for the title bar.

The reasons to customize the About window include to provide:

- ▶ Such information as telephone numbers and names of Customer Support for your company
- ▶ A seamless look to the products that are used by your company

The About window can be customized to contain company specific information such as the company name, and company logo.

The information that is displayed in the About window is obtained from a text file named product.inf. The file can be created using a text editor such as Notepad. The product.inf must be located in the OnDemand installation directory. The default installation directory is C:\Program Files\IBM\OnDemand32. Example 15-4 shows the contents of a sample product.inf file.

Example 15-4 Contents of the product.inf file

```
[Product]
NAME=Bank Archive System
LOGO_FILE= C:\Program Files\IBM\OnDemand32\bank.bmp
ABOUT_TITLE=Bank Customer Support
ABOUT_LINE1=To contact a customer support representative call:
ABOUT_LINE2=Customer Support Hotline
ABOUT_LINE3=1-888-BBB-HELP
ABOUT_LINE4=Data Processing Center
ABOUT_LINE5=1-888-552-5392
ABOUT_LINE6=For Online help view the web pages:
ABOUT_LINE7=http://www.BBB.CSHotline.com
ABOUT_LINE8=http://www.BBB.DataCenter.com
```

The title bar of the OnDemand main window is customized with the name Bank Archive System from the NAME keyword in the product.inf file (Figure 15-28).



Figure 15-28 Customized About menu

Figure 15-29 shows the customized About window using the sample product.inf shown in Example 15-4 on page 512.

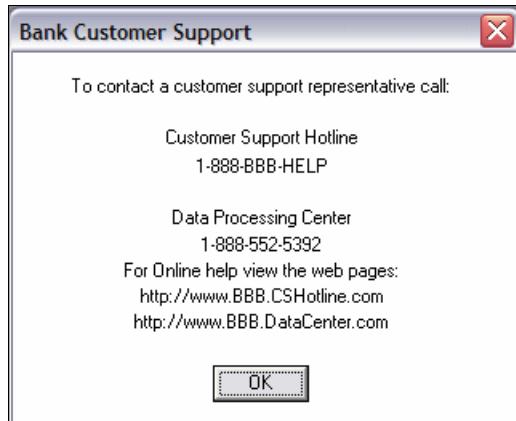


Figure 15-29 Customized About window

You can also customize how long you want to display the About window through the registry setting. Refer to 15.11.4, “Displaying the OnDemand splash screen or About window” on page 519, for more details.

15.11 Modifying client behavior through the registry

Many aspects of client operation can be modified only through entries in the Windows registry. Some of the more useful registry changes are described in this section. For information about other registry changes, see *IBM Content Manager OnDemand - Windows Client Customization Guide and Reference*, SC27-0837.

All of the registry entries are created in the key
HKEY_CURRENT_USER\Software\IBM\OnDemand32\Client\Preferences.

15.11.1 Single-selection hit list

Problem: The Document List (also known as the *hit list*) on the Folder window allows multiple documents (rows) to be selected simultaneously. This permits several documents to be retrieved with a single click of the View All Selected button. An administrator might want to remove this ability and restrict the selection to a single document, thereby permitting only a single document to be retrieved from the server at one time. This option can be used to control document retrieval load on the server.

Solution: For fix pack 7.1.2.4, a registry entry has been added to support this requirement.

Create a string value named SINGLE_SEL_DOCLIST. If assigned a value of 1 (Figure 15-30), the Document List allows only a single document to be selected. If a document is already selected and a new selection is made, the previously-selected document is deselected.

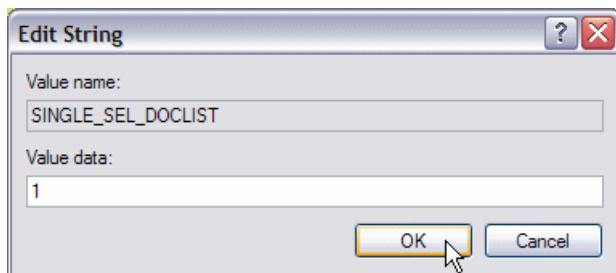
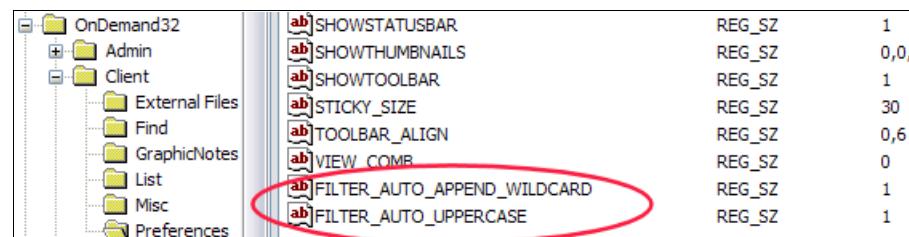


Figure 15-30 Registry setting modification: single-selection hit list

15.11.2 Folder List Filter enhanced

Problem: The Folder List Filter is used to create a subset list of all folders that available on a server. Users enter folder names and optional wildcard characters to specify the desired subset. Certain users require that a wildcard character be automatically appended to all folder names or that the folder names be automatically converted to uppercase when submitted to the server.

Solution: For fix pack 7.1.2.3, several registry entries have been added to support this requirement (Figure 15-31).



OnDemand32	SHOWSTATUSBAR	REG_SZ	1
Admin	SHOWTHUMBNAILS	REG_SZ	0,0,
Client	SHOWTOOLBAR	REG_SZ	1
External Files	STICKY_SIZE	REG_SZ	30
Find	TOOLBAR_ALIGN	REG_SZ	0,6
GraphicNotes	VIEW_COMB	REG_SZ	0
List	FILTER_AUTO_APPEND_WILDCARD	REG_SZ	1
Misc	FILTER_AUTO_UPPERCASE	REG_SZ	1
Preferences			

Figure 15-31 Registry setting modification: enhanced Folder List Filter option

Create a string value named FILTER_AUTO_APPEND_WILDCARD. If assigned a value of 1 (Figure 15-32), a wildcard character is automatically appended to each folder name submitted by the Folder List Filter.

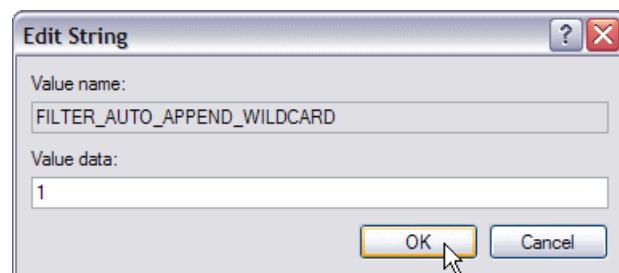


Figure 15-32 Registry setting modification: FILTER_AUTO_APPEND_WILDCARD

Create a string value named FILTER_AUTO_UPPERCASE. If assigned a value of 1 (Figure 15-33), each folder name submitted by the Folder List Filter is automatically converted to uppercase before it is sent to the server.

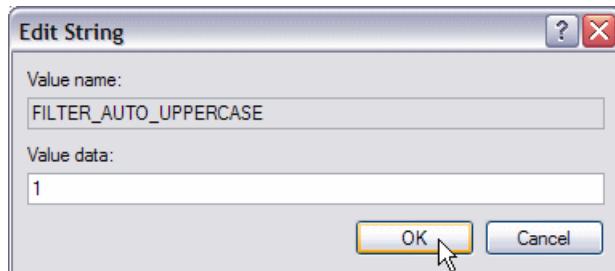


Figure 15-33 Registry setting modification: FILTER_AUTO_UPPERCASE

How this works in the OnDemand client

In our example, after setting both registry entries to a value of 1, entering the filter of pay results in a folder list of all folders starting with PAY, as shown in Figure 15-34.



Figure 15-34 Registry setting modification: Folder List Filter changes

15.11.3 Customizing your line data background

In the past, printers often printed on green bar paper (paper that alternates green stripes with white stripes) to enable users to better view listings and large accounting reports. The green bar background can be selected in the client or be set as the default by the administrator (Figure 15-35).

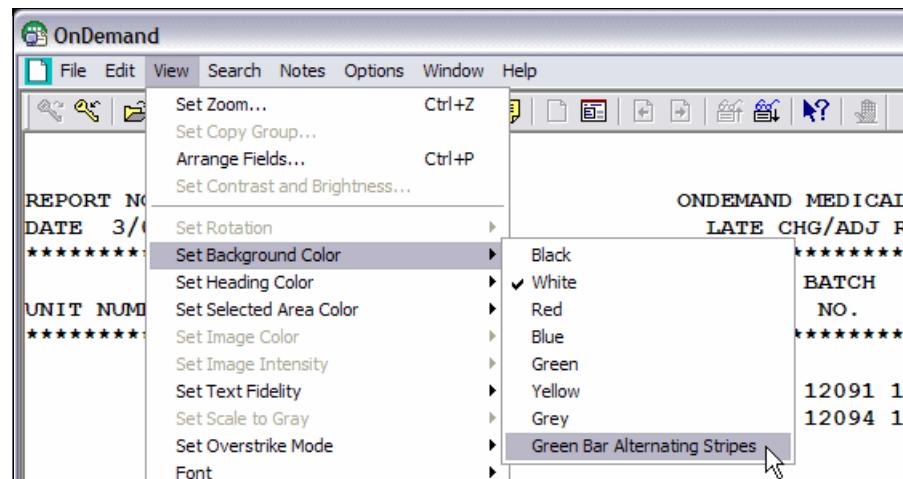


Figure 15-35 Line data background customization: Green Bar Alternating Stripes

The easiest way to determine the Red Green Blue (RGB) values is to use an application such as Microsoft Paint. From the menu bar, click **Colors** → **Edit Colors** → **Define Custom Colors**. Move the cross hairs and color slider until you find a pleasing color (Figure 15-36). Note the RGB values and enter them in the Windows registry entry.

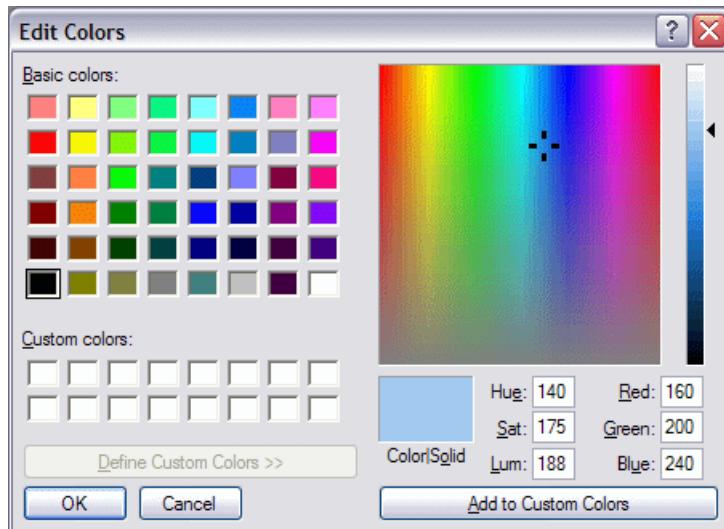


Figure 15-36 Editing colors

Create a string value named GREEN_BAR_COLOR. The value assigned overrides the color of the green bands of the green bar background color. It must be specified as a RGB value. For example, to use light gray bars instead of green, specify 230,230,230 (Figure 15-37). The default value is 128,255,128.

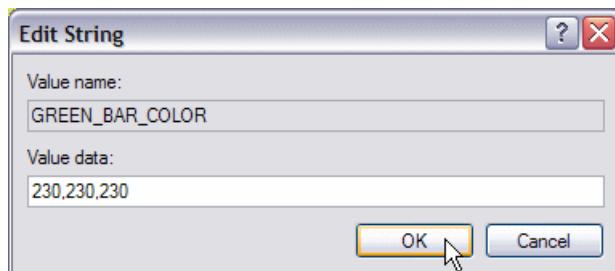


Figure 15-37 Registry setting: GREEN_BAR_COLOR

Figure 15-38 shows the result of this registry entry.

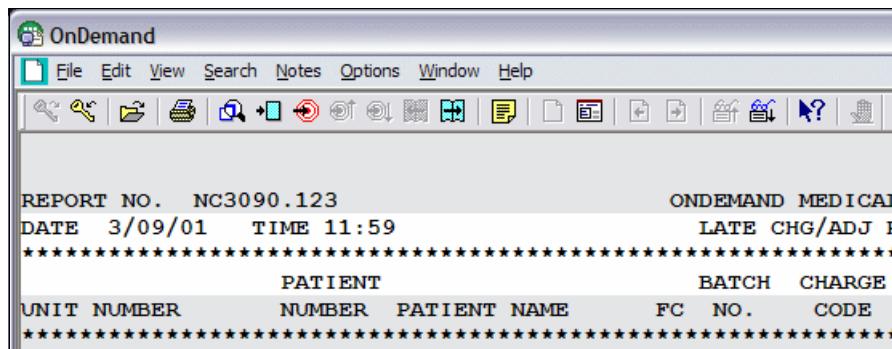


Figure 15-38 Output of registry setting for GREEN_BAR_COLOR

15.11.4 Displaying the OnDemand splash screen or About window

When the OnDemand client is first started, an OnDemand splash screen or an About window is displayed for approximately two seconds. By default, the OnDemand splash screen is displayed. However, if the About window has been customized or if the OnDemand splash screen bitmap file, ODSplash.bmp, does not exist in the OnDemand installation directory, the About window opens.

The amount of time to display the splash screen or the About window can be changed to a longer or shorter time by adding an entry in the Windows Registry. The display time is specified in seconds. A value of zero can be specified to prevent the splash screen or the About window from being displayed.

If you customized the About window to provide Customer Support information, it might be desirable to increase the display time. Alternatively, to provide a uniform look for all of the products used by a company, it might be desirable not to display the OnDemand splash screen so that the OnDemand client appears to be part of a suite of programs used by the company. For more information about customizing the About window, see 15.10, “Customizing the About window” on page 511.

Create a string value named SHOWLOGO. Assign a value of zero or more seconds. Figure 15-39 shows a registry file to set a display time of five.

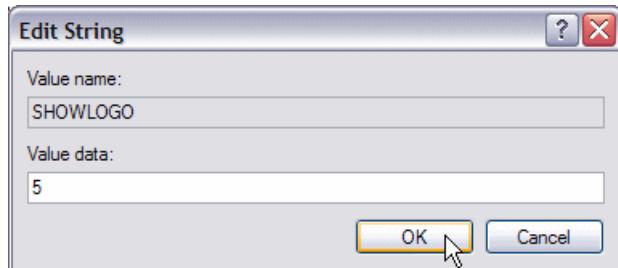


Figure 15-39 Registry setting: SHOW LOGO

After you create this registry entry, the OnDemand splash screen or the About window is displayed for 5 seconds when the OnDemand Client is started.

15.12 Negative numbers in decimal fields handling

A decimal number can have a leading negative sign (-13.75) or trailing negative sign (13.75-). The software that creates the report determines the position of the negative sign. The position does not change the meaning of the negative sign.

OnDemand processes a *leading negative sign* without any special steps in the definition of the indexer parameters. Simply define the length of the field long enough to include the negative sign and the largest possible value the field can contain. For example, if the largest possible value is -9,999,999.99, you define 13 as the field length. In the load information, OnDemand removes the leading and trailing blanks, embedded blanks, commas, and periods.

In Example 15-5, Field 3 (FIELD3) and Index 3 (INDEX3) contain the decimal numbers.

Example 15-5 Indexer parameters

```
FIELD3=0,72,13,(TRIGGER=3,BASE=0)
INDEX3=X'819496A495A3',FIELD3,(TYPE=GROUP,BREAK=N0) /* amount */
```

By default, OnDemand cannot process a trailing negative sign. The **arslog** command logs error 88 in the system log with text similar to the following example:

Row 1: The value '13.75-' cannot be converted to a valid decimal number.

There are special steps you can make when you define the indexer parameters to enable OnDemand to process the *trailing negative signs*. In summary, you must create two fields in the indexer parameters and use these fields to move the negative sign from a trailing position to a leading position.

Here are the special steps:

1. Define two fields. One field contains the numeric portion of the amount. The other field contains the sign portion of the amount.
2. Concatenate the two fields in the index definition, placing the sign portion first, followed by the numeric portion.
3. In the load information, remove leading and trailing blanks, embedded blanks, commas, and periods.

In our example as shown in Figure 15-40, Field 3 contains the numeric portion, and Field 4 contains the sign portion.

The screenshot shows a graphical indexer interface. At the top, it says "Trigger 3". Below that, there is a table with three columns. The first column contains the date "11/18/02" and the account number "3901234". The second column contains the name "BEN, BROOKE". The third column contains the amount "9,999,999.99". Arrows point from the text "Field 3" to the account number and from the text "Field 4" to the amount. At the bottom of the interface, there are buttons for "Ready", "Display", and "P".

11/18/02	3901234	BEN, BROOKE
Field 3	9,999,999.99	Field 4

Figure 15-40 Negative number capture in graphical indexer

Index 3 contains the decimal amount with Field 4 (FIELD4), the sign portion first, and Field 3 (FIELD3). See the resulting indexer parameters set up in Example 15-6.

Example 15-6 Indexer parameters

```
FIELD3=0,73,12,(TRIGGER=3,BASE=0)
FIELD4=0,85,1,(TRIGGER=3,BASE=0)
INDEX3=X'819496A495A3',FIELD4,FIELD3,(TYPE=GROUP,BREAK=N0) /* amount */
```

From an OnDemand client, the document list displays the amount with a leading negative sign. See Figure 15-41.

The screenshot shows an OnDemand client interface with a "Document List" header. Below it is a table with four columns: "Name", "Account #", and "Amount". The first row shows "BEN, BROOKE" in the Name column, "3901234" in the Account # column, and "-9999999.99" in the Amount column. The "Amount" column has a right-aligned dollar sign icon.

Name	Account #	Amount
BEN, BROOKE	3901234	-9999999.99

Figure 15-41 Negative number displayed in OnDemand client

15.13 Message of the day

The Message of the day is an easy way to inform users about:

- ▶ Server upgrades
- ▶ Education sessions
- ▶ New functions that are available
- ▶ Special events
- ▶ Other important information that people should know

Figure 15-42 shows an example of the message of the day.

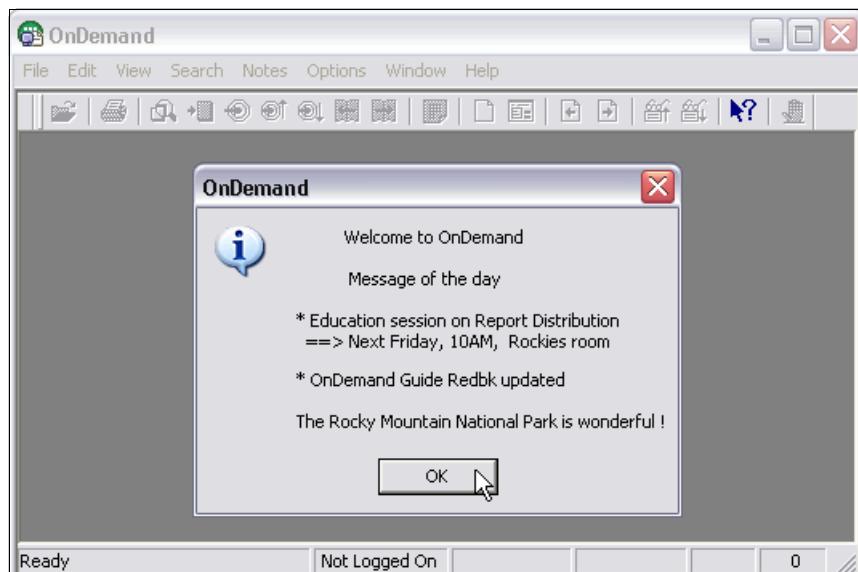


Figure 15-42 Message of the day

The content of the message file can contain a maximum of 1024 characters of text. The administrative client and the user client show the message after users log on to the server. To close the message box and continue, users click OK.

To set up the message of the day, choose one of the following options:

- ▶ For all OnDemand server platforms except Windows, set the ARS_MESSAGE_OF_THE_DAY parameter to the full path name of a file that contains the message that you want the client to show, in the ARS.CFG file, for example:

```
ARS_MESSAGE_OF_THE_DAY=/opt/ondemand/tmp/message.txt
```

- ▶ For a Windows OnDemand platform, add a String Value in the Windows registry. The String Value name is ARS_MESSAGE_OF_THE_DAY. Set the value to the full path name of a file that contains the message that you want the client to show. For example, see Figure 15-43.

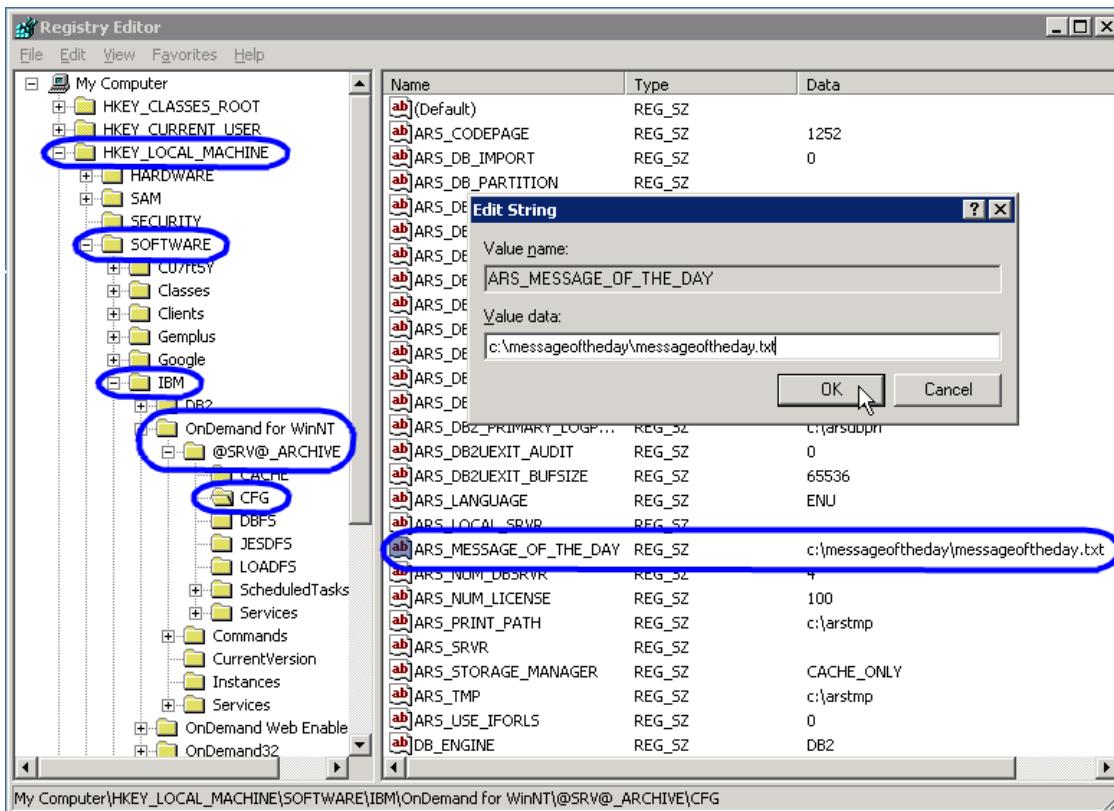


Figure 15-43 Windows registry for Message of the day

Restart the server after you modify the message of the day information.

15.14 OnDemand bulletins

IBM periodically distributes e-mail bulletins with tips, techniques, announcements, and product news related to the OnDemand for iSeries product. Since many of the technical tips also apply to other OnDemand platforms, you might want to subscribe to the bulletin even if you do not work with the iSeries server.

Go to the OnDemand for iSeries Support Web site at the following address:

<http://www.ibm.com/software/data/ondemand/400/support.html>

Search on the word bulletin. You can obtain summary bulletins from the last several years. Review them to find such valuable information as:

- ▶ Common problems and solutions
- ▶ Indexing techniques
- ▶ Client command line parameters
- ▶ Enhancements to the end-user client
- ▶ Enhancements to the administrator client
- ▶ ODWEK enhancements
- ▶ How to create an AFP overlay
- ▶ Tips on migration from Spool File Archive to Common Server
- ▶ OnDemand client upgrade considerations
- ▶ How to set up Document Audit Facility
- ▶ Tips on using query restrictions
- ▶ How to use Expression® Find in the client
- ▶ How to add your own messages to the System Log
- ▶ How to display a “message of the day” to an OnDemand client user
- ▶ How to use a public named query with arsdoc to make it easier to delete individual documents or modify index values
- ▶ How to use a folder list filter in the OnDemand client
- ▶ And many more tips

If you want to subscribe to the bulletin, contact Darrell Bryant by sending e-mail to dbryant@us.ibm.com.



Optional features

In this chapter, we explore optional features for Content Manager OnDemand system for different platforms.

We cover the following topics:

- ▶ OnDemand Distribution Facility (ODF) on z/OS
- ▶ Report Distribution
- ▶ Content Manager OnDemand Toolbox
- ▶ E-mail Notification and Delivery for Multiplatforms

16.1 OnDemand Distribution Facility (ODF) on z/OS

OnDemand Distribution Facility is the report distribution feature for IBM Content Manager OnDemand for z/OS V8.4. ODF is designed to group archived report pages or segments into print bundles for distribution.

The report distribution facility consists of:

- ▶ A started task
- ▶ A z/OS batch capture interface
- ▶ CICS Administration Client
- ▶ Monitoring facilities
- ▶ A batch utility

ODF obtains information from DB2 tables that can then be set up and maintained through an online administration facility. This allows generated bundles of captured reports for each user to be organized into print bundles for distribution.

The started task consists of a scheduler, which automates scheduling of report distributions. The scheduler includes the following processors:

- ▶ Continuation processor: Provides continued distributions for selected distributions.
- ▶ Distribution processor: Initiates available distributions.
- ▶ Print processor: Prints the bundles to be distributed.

The print processor may write the requested report pages or report segments to the JES spool with the appropriate delivery information, to an output z/OS data set, or to an e-mail URL. The print processor may also send a notice to an e-mail URL that a report has been output to a JES spool. For JES spool, multiple reports for a single user at a specific destination are combined into one bundle, or for each designated user into print bundles for distribution.

For more information about installing the optional ODF feature, refer to the *IBM Content Manager OnDemand Distribution Facility Installation and Reference Guide*, SC27-1377.

16.1.1 Additional features

OnDemand Distribution Facility offers the following additional features and functions:

- ▶ Reference field controlled distribution.

This feature permits control of what reports will be distributed. The feature uses a batch process to create the print requests for distributions of only the selected documents. It eliminates the overhead because ODF does not attempt to get a document that does not exist. This is a useful option when very large distributions have a small percentage of hits on the report to be distributed.

- ▶ E-mail server specification.

You can now specify the e-mail server in the Writer field of the Distribution Control Table (DCT) or the Bundle Definition Table (BDT). For further information about how to set entries in either the DCT or BCT, refer to the *IBM Content Manager OnDemand Distribution Facility Installation and Reference Guide*, SC27-1377.

- ▶ Ability to reprint an entire distribution.

The panel window (PL) now enables you to request a full distribution reprint.

- ▶ DB2 connection failure enhancement for the batch utilities.

The batch function and utilities have been enhanced to provide more meaningful information if a failure to connect to DB2 occurs at initialization.

- ▶ Tokenized queries and `query rebuild` command.

ODF now creates fully tokenized queries while accessing an OnDemand server. ODF supports both tokenized and non-tokenized queries. Queries are not automatically converted to tokenized queries. A new command has been added to the batch administration utility that allows you to rebuild some or all queries.

- ▶ ODF performance enhancements.

ODF now maintains a persistent connection under certain circumstances to the OnDemand archive server. This eliminates the logon and logoff processes for each server request.

- ▶ Bypassing TCP/IP usage.

You can set up ODF to invoke a local copy of the OnDemand archive server without using TCP/IP. This expedites access to the archived data and greatly improves the throughput of ODF.

- ▶ Persistent started jobs.

You can set up ODF to keep the started jobs active for as long as new distributions are available to be processed.
- ▶ Pre-check document existence.

This feature allows you to check documents for their existence before a distribution is set up for printing. It is only useful when there is a small number of distributions to be processed.
- ▶ Batch distribution processor.

A user-defined process can invoke the batch distribution processor to create the distribution status entries. It is useful where there is a large number of very large distributions. The batch distribution processor pre-checks the distribution for existing documents and sets up a print request for only the existing documents.

16.1.2 ODF administration process

In this section, you can administer ODF using the ODF monitor and administration screen.

ODF monitor

The ODF Monitor screen is an online, menu-driven monitor that enables you to perform the following tasks within ODF:

- ▶ Distribution inquiry
- ▶ Report bundle inquiry
- ▶ Recipient inquiry
- ▶ Destination printer inquiry
- ▶ Initiate distribution reprint
- ▶ Reprint inquiry

Figure 16-1 shows the main ODF Monitor screen with the Report Distribution Monitor Options.

```
OnDemand for z/OS and OS/390
OnDemand Distribution Facility

Option> _          Report Distribution Monitor Options:

Distribution Name for options 1 - 4:
Recipient/List      >
Distribution Description>
Search start date   > YYMMDDHRMN    end date  > YYMMDDHRMN
1 Distribution Inquiry
2 Requested Distribution Inquiry
3 Report Availability Inquiry
4 Report Reprint Inquiry
5 Report Inquiry      Report Name      >
6 Recipient Inquiry   Recipient      >
7 Destination Inquiry Destination     >
M Report Distribution Administration

Enter=Perform Selected Option   F1=Help   F3=End   Clear=Exit
```

Figure 16-1 ODF Monitor

ODF administration

The ODF Administration screen presents the administrator with a variety of options for administering report distribution. They are:

- ▶ Display and maintain recipients and recipient lists.
- ▶ Display and maintain distributions.
- ▶ Maintain report cross references.

Figure 16-2 shows the main ODF Administration Report Distribution screen with the above options.

```
OnDemand Distribution Facility
Report Distribution Administration

Option>

Recipient      >
1 Display Recipients
2 Maintain Recipients (Create/Delete/Update/Retrieve)

Recipient List  >
3 Display Recipient Lists
4 Maintain Recipient Lists (Create/Delete/Update/Retrieve)

Distribution Recipient/List>
Distribution Description  >
Job Name        >
5 Display Distributions
6 Maintain Distribution (Create/Delete/Update/Retrieve)

Multi-Platform Report Cross Reference
ReportID>
7 Maintain Report Cross Reference

Enter=Perform Selected Function   F1=Help   F3=End   Clear=Exit
```

Figure 16-2 ODF Administration screen

To maintain a recipient list, choose option 4 on the ODF Administration screen. This will present you with the screen shown in Figure 16-3.

```
UM          OnDemand Distribution Facility
            Maintain Recipient
Action Code > (CREATE, RETrieve, UPDate, DElete, CANcel)
Recipient>   Banner>      Y/N
Header 1    >
Header 2    >
Header 3    >
Header 4    >
Header 5    >
Header 6    >
Header 7    >
Header 8    >
Account     >
Address 1   >
Address 2   >
Address 3   >
Address 4   >
Building    >
Department  >
Name        >
Room        >
Title       >
ARS04014I NO MATCHING USER ID FOUND
Enter=Action F1=Help F2>Add to Recipient List>           F3=End
```

Figure 16-3 Maintain Recipient screen

The Maintain Recipient screen displays the following fields:

- ▶ Recipient field: Contains the user ID of the recipient.
- ▶ Banner field: Contains either a Y (create banner) or N (do not create banner).
- ▶ Header lines 1-8 fields: Contain special instructions as how to print output to the header pages of sysout. 1 to 60 characters are limited per line.
- ▶ Account field: Contains accounting code information for this recipient. The output to an ODF header page of a bundle is limited to 1 to 55 characters.
- ▶ Address lines 1-4 fields: Contains the user address information to be print out to an ODF header page of a bundle. This is limited between 1 to 60 characters per line.
- ▶ Building field: Contains building identification to be printed out (output) to an ODF header page of a bundle. This is limited to between 1 to 60 characters.
- ▶ Department field: Contains the department identification to be output to an ODF header page of a bundle. This is limited to between 1 to 60 characters.
- ▶ Name field: Contains the recipient name to be output to an ODF header page of a bundle. This is limited to between 1 to 60 characters.
- ▶ Room field: Contains room identification to be output to the separating page of a bundle. This is limited to between 1 to 60 characters.
- ▶ Title field: Contains title information to be output to the separating page of a bundle. This is limited to between 1 to 60 characters.

To display a recipient's information, select Option 1 from the Report Distribution Administration screen. See the recipient field data shown in Figure 16-4.

```
UM - OnDemand Distribution Facility
      Maintain Recipient
Action Code > (CREATE, RETrieve, UPDate, DElete, CANCEL)
Recipient> ODFUSR1 Banner> Y/N
Header 1 >
Header 2 >
Header 3 >
Header 4 >
Header 5 >
Header 6 >
Header 7 >
Header 8 >
Account >
Address 1 > ODF CORP A
Address 2 > 101 OAKS DRIVE
Address 3 > CHARLESON, CA
Address 4 >
Building > 1801
Department > IT
Name > BILL SMITH
Room >
Title >
ARS040131 MATCHING DATA FOUND
Enter=Action F1=Help F2>Add to Recipient List> F3=End
```

Figure 16-4 Maintain recipient list screen - Display recipient information

To display a list of recipients from the Report Distribution screen, select Option 1 to display the recipients. See the displayed list of the recipients defined in the ODF database, as shown in Figure 16-5.

UL - OnDemand Distribution Facility
Display Recipients Page
Recipient> _____ Change search field to generate new 1
Recipient **N**ame/**D**escription
 ODFUSR1 BILL SMITH
 ODFUSR2 BILL JONES
 ODFUSR3 JANE JONES

ARS04006I LAST DETAIL LINE ON THIS SCREEN IS END OF LIST
Action options - S=Select C=Copy As Model /=Reposition Item
Enter=Action F1=Help F3=End F7=Pg Bkwd F8=Pg Fwd CLEAR=Main

Figure 16-5 Display Recipient: Display recipients defined in ODF database

To access the Maintain Cross Reference Table screen, select Option 7 from the Report Distribution screen and this will take you to the Cross Reference Maintenance screen, as shown in Figure 16-6.

OnDemand Distribution Facility Multi-Platform Cross Reference Maintenance				Page 001 of
S	ReportID	Application Groups	Applications	Reference
-	AFP	A afpdata	afpdata	
-	AFPBILL	A afpdatai	afpdatai	
-	AFPDATA	I afpdata	afpdata	NP
-	AFPDATAB	I afpdatab	afpdatab	
-	AFPDATAI	A afpdatai	afpdatai	
-	AFPDATA2	I afpdata	afpdata	NP
-	AFP2	I afpdata	afpdata	NP
-	AFTST	A AFTST	AFTST	
-	AFTSTG	A AFTST	AFTST	
-	AFTSTZ	A AFTST	AFTST	
-	ALT1	A ALT1	ALT1	
-	APKIVP	A apkivp	apkivp	
-	BASE01	A afpdata	afpdata	NP
-	BASE02	A afpdatab	afpdatab	
-	BASE03	A CHKSTMNT	CHKSTMNT	
-	BASE04	A CHK	CHK	140172594

ARS040121 UPDATE SUCCESSFUL
Action options - A=Add C=Copy D=Delete E>Edit U=Update R=Rename /=Re-posi
Enter=Search/Action F1=Help F3=End F4=Refresh F7=Pg Bkwd F8=Pg Fwd

Figure 16-6 Cross Reference Maintenance screen

The Cross Reference Maintenance screen shows the following information:

- ▶ Report ID (1-8) field: Used to track the document as it is processed for distribution.
- ▶ Status field (1): Contains identification information, and the status of the report on OnDemand V7; A is for active report and I for inactive report.
- ▶ Reference fields: Can be used to control when a report is available for distribution. This is used in conjunction with a marked index column in OnDemand. If there is a match, ODF will match the Reference value to a column index value and set the report for distribution.
- ▶ Index fields: Index Column Name is used when the reference caching option is used. This column identifies the name of the index column specified in the queries for segment selection. It must match the index column name exactly. It becomes part of the HFS directory name where the segment files are cached for quick retrieval.
- ▶ Application Group Name fields (1 - 60): Must match an existing one defined within the OnDemand database.
- ▶ Application Name fields (1 - 60): Must match an existing one defined within the OnDemand database.
- ▶ Host Name field (1 - 120): Contains the name of the URL or IP address of the OnDemand Application host server.

Defining the Distribution Control Table (DCT)

The DCT is a DB2 table. It contains control and default information for the processing of the distribution. It contains the following information:

- ▶ Distribution Name (Recipient and Description)
- ▶ Accounting information
- ▶ Distribution Initiation Method
- ▶ Job Name information
- ▶ Continuation processing information
- ▶ Manifest Report flag
- ▶ Report Break information
- ▶ Print Parameters

To maintain distribution within ODF, select Option 5. See Figure 16-7.

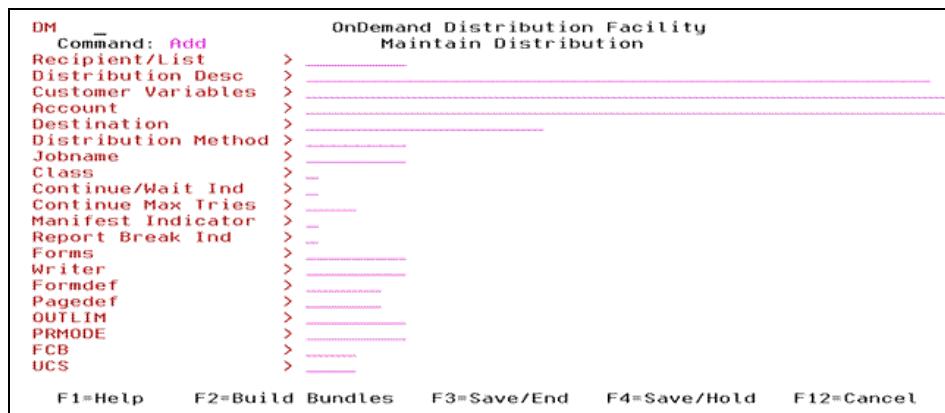


Figure 16-7 Maintain Distribution screen - With option 5

In the Maintain Distribution screen, you can update the following information:

- ▶ Recipient and Distribution fields: Descriptions for both fields together make up the Distribution Name.
- ▶ Recipient/List field: Contains the user ID in the User Output Table (UOT) or the Recipient List in the Recipient List Table (LIS).
- ▶ Distribution Description field: Contains a description indicating characteristics of the distribution, such as purpose, contents, or recipients.
- ▶ Customer Variables field: Contains sysout parameters used to override sysout parameters prior to dynamic allocation using the pre-allocation exit.
- ▶ Account field: Contains job card accounting code information for this distribution.

- ▶ Destination field: Contains the output destination printer name. In addition, a system node name may be included.
- ▶ Distribution Method field: Contains ALLREADY, LOADED, EXTERNAL, TODHH:MM, TOPHH:MM, and TOSHH:MM.
- ▶ Job Name field: Contains the Job Name used for print processors created from this distribution used for unique identification in JES.
- ▶ Class field: Contains the JES SYSOUT class. The default class for use when sending the distribution to the printer.
- ▶ Continuation/Wait field: Contains C (for continuation processing) or W (to wait for all reports within distribution before processing).
- ▶ Continuation Max tries field: Contains the number of times a bundle should go through continuation processing for this bundle. The number should not exceed the total number of continuation intervals from the scheduled time to midnight of current day.
- ▶ Manifest Indicator field: Contains Y (to generate a manifest report) or N (not to generate a manifest report).
- ▶ Report Break Indicator field: Contains Y (to generate multiple sysout) or N (to generate a single sysout for the distribution bundle).
- ▶ Other fields: Contain standard print parameters.

Selecting Option 6 to access the Maintain Distribution within ODF will take you to the screen shown in Figure 16-8.

OnDemand Distribution Facility	
Maintain Distribution	
DM	Command:
Recipient/List	> <u>ODFLIST</u>
Distribution Desc	> <u>ODFDIST12</u>
Customer Variables	> _____
Account	> _____
Destination	> <u>PRT12</u>
Distribution Method	> <u>ALLREADY</u>
Jobname	> <u>RDAR22A</u>
Class	> <u>H</u>
Continue/Wait Ind	> <u>W</u>
Continue Max Tries	> <u>0001</u>
Manifest Indicator	> <u>Y</u>
Report Break Ind	> <u>N</u>
Forms	> _____
Writer	> _____
Formdef	> _____
Pagedef	> _____
OUTLIM	> <u>00000000</u>
PRMODE	> _____
FCB	> _____
UCS	> _____

F1=Help F2=Build Bundles F3=Save/End F4=Save/Hold F12=Cancel

Figure 16-8 Maintain Distribution screen - With option 6

To display a distribution list, select Option 5 from the Report Distribution Administration screen and this will take you to the screen shown in Figure 16-9.

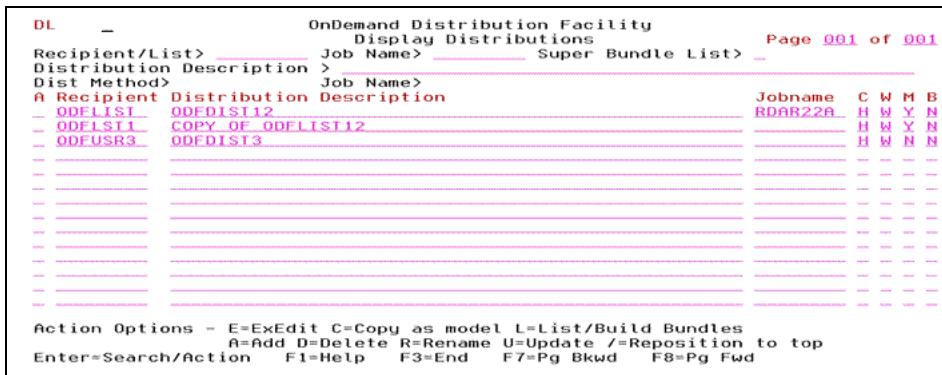


Figure 16-9 Display Distributions screen

From the Display Distributions screen, administrators have the ability to perform the following tasks:

- ▶ To display the list of distributions that are defined in the ODF database.
 - ▶ The ability to perform maintenance against some of the distribution fields.
 - ▶ The ability to:
 - ExEdit: Edit the existing distribution information.
 - Copy as a Model: To COPY a distribution using the contents of the existing distribution you are copying from, including all bundles.
 - List of Bundles: Lists bundles for the distribution you selected.
 - Add: Add a distribution.
 - Delete: Delete a distribution.
 - Rename: Rename a distribution with a new Recipient/List name and Description.
 - Update: Update Job Name, class, wait, continue, manifest, and banner indicators.

Defining the Bundle Definition Table (BDT)

The BDT is a DB2 table that contains report information defining the contents of a distribution. This report information consists of the following:

- ▶ Sequence
 - ▶ Report ID
 - ▶ Version
 - ▶ Status

- ▶ Print Class
 - ▶ Wait/Ignore Processing Information
 - ▶ Report selection criteria

The Display Bundle Definition screen is shown in Figure 16-10.

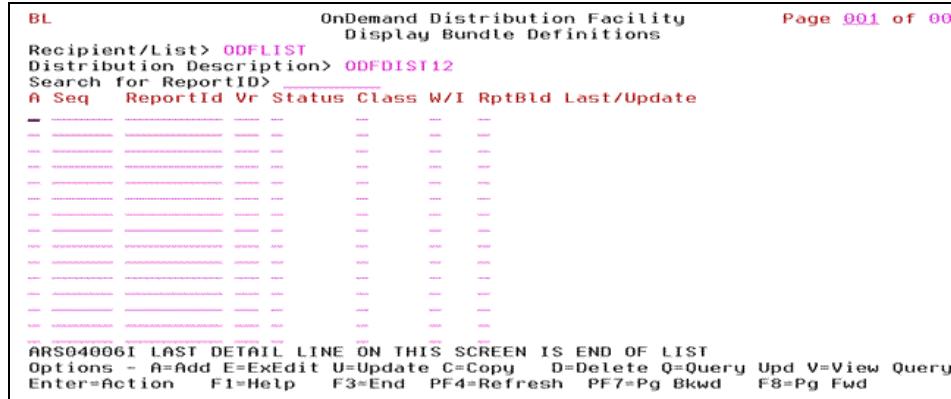


Figure 16-10 Display Bundle Definition screen

The Display Bundle Definition screen displays the following information:

- ▶ Sequence field: Controls the order of all the reports defined within a distribution.
 - ▶ Report ID field: The name of the report defined in the Cross Reference Table.
 - ▶ Version field: The version of the report defined in the OnDemand database. The default is 01 for OnDemand V7. For OnDemand V2, the default will always be the latest version of the report.
 - ▶ Status field: The status of the report with the distribution: A is active and I is inactive.
 - ▶ Class field: The default output class used when sending the report to a printer.
 - ▶ Wait/Ignore field: Contains W (to indicate that the distribution should wait on this report if it is unavailable at distribution initiation time) or I (to indicate that the distribution should not wait on this report).
 - ▶ Report Build indicator: Contains F if the entire report is to be distributed or Q if the report is to be distributed by data selection range.

Figure 16-11 shows a bundle definition with predefined fields containing information.

Figure 16-11 Display Bundle Definition screen - with predefined fields

The Maintain Bundle Definition Report screen contains the following fields:

- The Report ID and Report Version fields: Contains the OnDemand for z/OS report name and version to be included in the distribution, as defined in the CRT table (OnDemand V7 only). The version will always be 01 for OnDemand V7. The Report Version may be ** or the report version number in OnDemand V2.
 - Customer Variables field: Contains sysout parameters used to override sysout parameters prior to dynamic allocation using the Pre-Allocation exit.
 - Destination field: Contains the output destination printer name for this report. A system node name may be included.
 - Job Name field: Contains the Job Name used for the print processor created for this report used for unique identification in JES.
 - Bundle Sequence field: Contains a unique sequence number assigned for a particular occurrence of a report within a bundle.
 - Report Build field: Contains F if the entire report is to be distributed, or Q if the report is to be distributed by page or data selection range.
 - Status field: Contains A to mark this bundle Active, or I to mark this bundle Inactive.
 - Wait/Ignore field: Contains W to indicate distribution should wait on this report if unavailable at distribution initiation time, or I to indicate distribution should not wait on this report.

- ▶ Location field: Contains one of the following characters:
 - S: For distribution to a printer.
 - E: For e-mail notification.
 - N: For a report sent to spool; notification of completion is done through e-mail.
 - D: For a report sent to an OS/390 dataset.
- ▶ Email Addr field: Contains the e-mail address to be used for e-mail notification or the dataset name when distributing to a dataset.
- ▶ Other fields: Contain standard print parameters.

An example of the Maintain Bundle Definitions screen is shown in Figure 16-12.

BM		OnDemand Distribution Facility	
		Maintain Bundle Definition	
Recipient/List	> ODELIST	Bundle Sequence	> 00010
Distribution Desc	> ODEDIST12		
Report ID	> CHRTIMN1	Report Build	> 0
Report Version	> **	Start Page	> 0000000
Status	> A	End Page	> 0000000
Class	> H	UCS	> _____
Wait/Ignore Ind	> W	COPYM	> _____
Destination	> PRI12	FCB	> _____
Jobname	> RDAR22A	TRC	> -
Writer	> _____	BURST	> -
Forms	> _____	CHAR1	> _____
FORMDEF	> _____	CHAR2	> _____
PAGEDEF	> _____	CHAR3	> _____
OUTLIM	> 00000000	CHAR4	> _____
Copies	> 001	FLASH	> _____
LOCATION	> S	PRMODE	> _____
Email/GDG (cont.)	> _____		
Customer Variables > _____			
Enter=Act F1=Help F3=Save/Return F6=Query F7=Prev F9=Get Default F12=Cancel F4=Save/Hold F8=Next			

Figure 16-12 Maintain Bundle Definitions screen sample

Bundle Definition Report: Date range search

Figure 16-13 on page 539 shows an example of a date range search. The fields in on this screen are:

- ▶ Sequence column: Determines the order of the query precedence. Placing a D in the column will delete the query.
- ▶ Negate column: Reverse or negate the logic of the query.
- ▶ The '(' column is used to logically group parts of a query.
- ▶ The 'L' column is used as the logical operator. Valid values are 'A', 'O', or space. There is also an And or Or built-in query.
- ▶ Field Name Key: Used to select the field name to be used in the comparison.

Figure 16-13 Bundle Definition Report Date Ranges screen

Distribution processing

The scheduler task begins the distribution process at distribution intervals using an activation routine. The scheduler and distribution intervals are defined by the DISTSLEEP and SCHDSLEEP system-wide parameters. The following are available distribution methods:

- ▶ ALLREADY
 - ▶ TODHH:MM
 - ▶ TOPHH:MM
 - ▶ TOSHH:MM
 - ▶ LOADED
 - ▶ EXTERNAL

Distribution processing checks to see if distributions are ready to print. It also has the capability to monitor the availability of captured reports. It notifies the main task that all of a distribution reports are available and ready to print.

Continuation processing monitors the ODF for z/OS work queues for missing reports and initiates a print when the reports are available.

Distribution method ALL READY

The scheduler examines distribution requests and starts the printing cycle for distributions defined with the ALLREADY distribution method when all reports defined to a distribution bundle are available.

ALLREADY distributions may be manually initiated even if all reports are not available.

If all reports defined to a distribution bundle are not available, the distribution request will be reexamined at the next distribution interval. If Continuation is defined for the distribution, the missing report(s) will be distributed when they are available.

Distribution method TODHH:MM

The Scheduler examines distribution requests and starts the printing cycle for distributions defined with the TODHH:MM distribution method when the time of day, as defined in hours and minutes, occurs.

TODHH:MM distributions may be manually initiated before the defined TOD occurs.

All available reports defined to a distribution bundle will be distributed. If Continuation is defined for the distribution, the missing report(s) will be distributed when they are available.

Distribution method TOPHH:MM

The scheduler examines distribution requests and starts the printing cycle for distributions defined with the TOPHH:MM distribution method when the bundle is complete or a partial distribution will be initiated when a specified time of day, as defined in hours and minutes, occurs.

TOPHH:MM distributions may be manually initiated before the defined TOP occurs.

All available reports defined to a distribution bundle will be distributed. If Continuation is defined for the distribution, the missing report(s) will be distributed when they are available.

Distribution method TOSHH:MM

The scheduler examines distribution requests when the CPU clock reaches the specified time of day. All distributions will be initiated within a super bundle at a specified time of day represented as hh for hours and mm for minutes. The super bundle distribution, and each of the distributions defined to the same super bundle, will share a unique Job Name value.

A main distribution is defined with:

- ▶ A distribution method of TOS:HH:MM: the time of day the super bundle job will start.
- ▶ A Job Name that uniquely defines the super bundle.

- ▶ The matching Job Name is the key to grouping a super bundle.
- ▶ All available reports defined to a distribution bundle will be distributed.

If Continuation is defined for the distribution, the missing report(s) will be distributed when they are available

Distribution method LOADED

The scheduler examines distribution requests and starts the printing cycle for distributions defined with the LOADED distribution method when all reports defined to a distribution bundle are available.

LOADED distributions may be automatically initiated even if all reports are not available by using the C indicator rather than the W. Often there is only one report for the LOADED distributions.

If all reports defined to a distribution bundle are not available, the distribution request will be reexamined at the next distribution interval. If Continuation is defined for the distribution, the missing report(s) will be distributed when they are available.

Distribution method EXTERNAL

The scheduler examines distribution requests and starts the printing cycle for distributions defined with the EXTERNAL distribution method when a print processor request is found in the Distribution Request Table (DRT).

Batch program ARSRDFGO is provided to add an entry to the DRT for EXTERNAL distributions.

Print processing

Print processing creates print bundles that consist of:

- ▶ A manifest page describing the contents of the print bundle, if requested
- ▶ A banner page preceding each report, if requested
- ▶ The entire report
- ▶ Selected page ranges of a report
- ▶ Selected documents (segments) of a report
- ▶ Print processor sysout under main task, ARSODF

Report Distribution Inquiry options

We describe the Report Distribution inquiry options as follows and what you can search on:

- ▶ Distribution Inquiry
 - Status by Distribution Name
 - Missing reports

- ▶ Requested Distribution Inquiry
 - Status by Distribution Name
 - Initiate distribution
- ▶ Report Availability Inquiry
 - Status by Report Name
- ▶ Report Reprint Inquiry
 - Reprint Status
 - Initiate Reprint
- ▶ Report Inquiry
 - Distribution definition by Report Name
- ▶ Recipient Inquiry
 - Distribution definition by Recipient Name
- ▶ Destination Inquiry
 - Distribution Status by Printer
 - Initiate reprint by Destination

Reprint Facility

Reprint the original distribution:

- ▶ Entire Distribution
- ▶ By Report name
- ▶ By Recipient/List name
- ▶ By Destination

It overrides the original print parameters and manages the Print Processor entries.

E-mail notification and delivery

Provides the ability to notify users that documents have been archived and are available for viewing. It also provides the ability to mail newly archived documents to users.

Figure 16-14 shows the maintain bundle definition screen with the option for e-mail notification and delivery set with E- or N- .

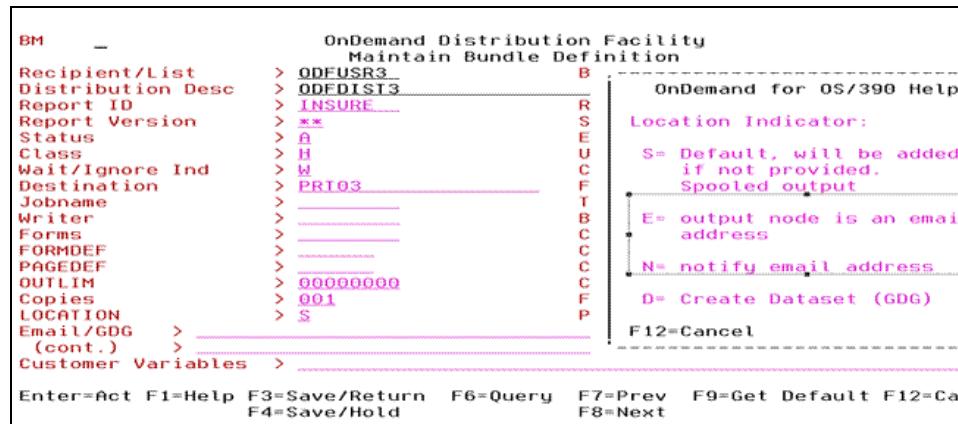


Figure 16-14 Maintain Bundle Definition: e-mail notification and delivery options

Distribution tables

The distribution tables in ODF and their descriptions are summarized in Table 16-1.

Table 16-1 ODF distribution tables

Distribution table	Description
Bundle Query Table (BQT)	Defines queries used to build the Bundle.
Print Query Table (PQT)	Defines the report query and the date of the query for the Distribution Bundle.
Recipient List Table (LIS)	Defines a list of recipients (user IDs) for print distributions.
Print Processor Table (PPT)	Defines printed distribution bundles used to produce initial print and reprint distribution output.
User Output Table (UOT)	Defines separator page and optional banner page header information for a print distribution recipient.
Cross Reference Table (CRT)	Contains a list of report names that cross-reference the ODF report name to the OnDemand application group/application name.

ODF documentation

For more information about the OnDemand Distribution Facility, refer to the *Installation and Reference Guide V7.1*, which can be found at the link below:

<http://www.ibm.com/software/data/ondemand/390/library.html>

16.1.3 Updated documentation and APAR lists

The ODF Reference Manual has been updated to include new features and functions in the current version of ODF. Documented APARs have been included in this section, which have been added to the manual since the release of ODF Version 7.

APAR 039140

This PTF applies to all the OnDemand Distribution Facility installations.

This PTF implements the unified logon capability when accessing the OnDemand Server. This eliminates the need to specify the user ID and password in the System Default table. The unified logon feature uses the TSO ID of the person who submitted the batch utility jobs or logged on to the CICS Administration Client Region. In this case, the user ID must be defined to the OnDemand Administration Client with the necessary authority.

Note: Do not remove the SDT entries that contain the user ID and password. These are the numbers 9, 24, and 25. These entries may be blanked out to utilize the unified logon feature of OnDemand.

APAR AD959240

This PTF adds a new feature to the OnDemand Distribution Facility. It permits the control of which designated reports will be distributed. Any given report may be generated by multiple jobs, all of which will be archived by OnDemand but not all of these will be necessarily valid for distribution. This is controlled by an assigned index value. The index value can be selected by checking the **Reference** check box found in the Application Group Field Information tab within the OnDemand Administration Client. The value of this reference field index is passed to ODF as a Reference field. A new field is then added to the ODF report Cross Reference table. This field is also referred to as a Reference field. If the value of the reference field sent to ODF matches the reference field of a table entry in the Cross Reference table, this report will be distributed to all defined recipients. The reference index field selected should be a single value index for a given application group; otherwise, the value that will be passed to ODF will be first value found for that index when the document was loaded into OnDemand.

For example, the report TRIALBAL is generated *ad hoc* throughout the day. Only a segment of the report is created each time. TRIALBAL has dozens of recipients defined in distributions to ODF, and each one is to receive only the final run.

Each *ad hoc* generation is run using a Job Name of DAYBAL. The final job that is run is called FINALBAL. OnDemand has an exit that can be applied to collect the Job Name and insert an index value into the TRIALBAL report. The TRIALBAL report has the JOBNNAME index value defined as a Reference field. When the DAYBAL job is run, the value DAYBAL is stored for the index of JOBNNAME.

When the FINALBAL run is executed, the index value stored is FINALBAL. When the ODF exit is invoked while the report is being archived, the index value for the Reference field is sent to it. The ODF Cross Reference table's Reference field has a value defined for the TRIALBAL report of FINALBAL. The ODF exit uses the value passed to it to match it against the Cross Reference table; when the DAYBAL value is used, no match is made, so nothing will execute. However, if the passed value is FINALBAL, ODF will find all recipients for this report and schedule the report for distribution.

An additional feature has been added to OnDemand Distribution Facility to control the scheduling of certain distributions. The Customer Variable field defined at either the Distribution or Bundle entry level can be coded with the words DO NOT SCHEDULE or NOSCHED. If either phrase is found at the bundle level, that bundle entry in a distribution will not be scheduled. If either phrase is found at the distribution level, then the entire distribution will not be scheduled. It is not dependant on what distribution method has been selected; the distributions that have been coded with this phrase may be scheduled externally. Once they have been scheduled, they will be processed as usual by the print processor.

ADD97728O

This technote introduces the new feature Reference Field Caching Option for OnDemand Distribution Facility. This feature adds the capability to pre-process documents for a distribution so that only documents that exist will be scheduled for printing. There are two methods of checking that this action is performed for existing documents. The first method requires the distribution method of EXTERNAL and will create a document cache in the Hierarchical File System to temporarily store the documents for fast retrieval. A set of criteria must be met before a document cache will be built. When a predetermined number of hits have been made, the document cache will be created. During print processing, the document cache will be checked first; if it is not found, the server will be queried. This method permits only simple queries to be used (queries that have a single compare value or IN a list of values). However, if a distribution that is intended to use the cache has a bundle entry that has a more complex query, the query will be used to check the server for a match. So both complex and simple queries can be specified. A Print request will only be inserted for documents that are found.

Using the document cache with simple queries will save significant processing, since the full document will be retrieved once from the server and temporarily stored for fast retrieval of the segments. This can have a huge benefit when there is a high percentage of hits within a document search.

The second method requires the distribution method of LOADED and involves checking the server directly. This method should be used when queries are more complex. However, a document cache will not be built. Each request will be checked directly against the server. This can save significant time during ODF print processing when large distributions do not have 100% hit ratio on existing documents, since the print requests are only generated for existing documents.

Both methods require the reference field to be set up on the CRT table and have the reference check box marked in the OnDemand Administration Client. Also, the cache keyword must be used as a load parameter, even though a document cache may not be created. This is because the cache keyword instructs the load to insert the request into the RIT table instead of the DST table.

ODF documentation updates

Full document requests are supported. However, if you are using the batch scheduler for a full document print, requests will not provide any improvement in throughput. It is feasible to combine full document and segment selection queries within distribution that will be scheduled using the batch scheduler. The intent of this option is to provide rapid distribution of large reports that are segmented into small documents that have many recipients.

How RFC operates

ODF may temporarily store all the documents of a report into a file cache. During print processing, ODF will first check if the document exists in the cache and will return it for printing if it exists. During load processing, a check will be made for the Reference index field. Any index field can be defined as a reference field using the Reference check box in the OnDemand Administration Client. When this check box is selected, ARSLOAD will call the ODF interface with the first value defined for this index. The ODF interface will check for the reference value in the Cross Reference table (ARSCRT). A match of the reference field will cause the interface to place an entry into the new table ARSRIT if the load has been invoked with the keyword CACHE. Without the CACHE keyword, normal scheduling of the defined distributions will be processed. The scheduling of the distributions associated with the document will be done by the batch scheduling process.

The batch scheduler will retrieve the ready ARSRIT entries. All distributions that have been defined to use the report will be retrieved. When the distribution method is EXTERNAL, those bundle entries that have been defined with a query will be checked first for a simple query, then a match on the index name in the CRT table will be checked first for a simple query, then a match on the index name in the CRT table, then finally a scan of the index data for this report will be done for the value defined in the query. A match will cause the cache to be built for this report once a threshold setting is reached. The threshold value allows for a certain number of hits before a cache will be built. Distributions may be scheduled for delivery prior to the cache being built. The threshold value provides control over when a cache will be built. If the number of distributions is small for a given report, setting the threshold above that number of distributions would prevent that cache from being built. The advantage of this is that a very large document to be cached may take longer to cache, but then it may be distributed normally. This threshold value also allows for some overlapping of printing distributions while the cache is being built. No recommendation is provided as to the best value for the threshold. The default value is 2, but can be set in the ARSSDT table to a value up to 9999. If a given distribution does not meet the cache requirements or is defined with a distribution method of LOADED, it will still be scheduled for regular processing. If a query is defined on the bundle entry, the query will be used to determine if a document exists. If no document exists, then a PPT entry will not be inserted. The report indicates which distributions have been scheduled for cache processing and which do not meet the criteria and also why they do not meet the necessary criteria.

Selection criteria for the RFC option

Selection criteria for RFC option includes the following:

- ▶ Reference check box set for index column

The Reference check box is intended to be used with a Job Name exit that sets the value of this index column to the Job Name that executes the OnDemand load; the value of this index column to the Job Name that executes the OnDemand load, however, may be used by any index column. Multiple jobs may generate a particular report that is loaded to the OnDemand archive, but only certain reports may be distributed by ODF. This is controlled by defining the corresponding reference field value in the Cross Reference Table Reference column.

- ▶ Cross Reference Table Reference field

The reference field must contain the value of the index column selected for the reference check box in the OnDemand Administration Client. If the reference column is a Job Name index, then the Job Name that generates the archived document to be distributed would be placed in this field.

► Index Field Name

The Cross Reference table now has the name of the index column that will be used for the segment selection defined here. This must match the column defined in the query selection statement. The index name specified here will not normally be identical to the reference index column.

► Cache Keyword

The cache keyword must be set in the load in order for the Cache feature to be used. Without the cache keyword, ODF will initiate distributions without using the RFC option.

► External Distribution Method

All distributions that are going to use the RFC option must be coded using the EXTERNAL distribution method to allow the document cache to be built. The LOADED distribution method can also be used, but no document cache will be built.

► Queried Report Build

The Report build indicator must be set to Q for a query selection. This is set in the bundles entries for the distribution.

► Simple queries

A query must be defined using the index column name specified on the Cross reference table. The query may only use the equal (=) or IN operators. The compare value must conform to the SQL rules for string values. That is a value that has been defined by quoted pairs. In the case of using the IN operator, the list of values must be surrounded by parenthesis and quoted pairs that are comma separated. More complex involved queries can be used when the distribution method is LOADED, and then the formed query must be stored into the ARSPQT table.

How to set up RFC for ODF

To set up RFC for ODF:

1. Set up Application Group Reference Field.
2. Sign on to the OnDemand Administration Client.
3. Select Application Groups.
4. Right click the Application group and select Update.
5. Select the Field Information tab.
6. Select the arrow in the Name drop-down box and click the index field that will be the reference field.
7. Click the Reference check box.

Set up the reference field and the Index column name:

1. Sign on to CICS.
2. Enter the ARON transaction.
3. Select option M on the Main ODF menu.
4. Select Option 7 on the Maintenance Menu.
5. Search for the report that will use the reference.
6. Select the report with an E to enter edit mode.
7. On the edit screen, find the Reference field. Enter the reference value. Five lines of reference field are available. The reference value may be up to 254 characters. It likely will only be a few, so enter the information to match the index column in the first line.
8. Enter the index column name for the specific segment selection in the Index Name field on the edit screen. This name is not necessarily the same as the reference field selected (more than likely it is not). This name is the same name used in the query for report segment selection.

To set up the CACHE keyword for ARSLOAD:

The -Z parameter now has three values that can be set. Possible values are -Z ODServer, -Z ODServer,CACHE or -Z ODServer,CACHE,TRACE. The trace keyword may be present as the second value if the CACHE key is not set.

To set up distribution settings:

1. Select the M option from the main ODF CICS screen.
2. Select 5 for the Distribution List.
3. Select or add the Distribution that will use the RFC option.
4. In the Distribution Method field, enter the EXTERNAL distribution method.
5. Press the F2 key to build a bundle entry.
6. Enter A for Add, the report ID entered in the CRT table, 01 for version number, and other values as appropriate. Finally, enter a report build (RptBld) value of Q.
7. Press Enter to add this entry, and then press the F6 key from the Maintain Bundle Definition screen to go to the query definition screen (B3).
8. Enter the Key number for the index field to be selected.
9. Enter a operator value of = or I (from IN).
10. If the = operator is used, enter the compare value surrounded by single quotes.

- 11.If the I operator is used, enter a list of values in the form ('value1','value2','value3'). The parenthesis and comma separators are required for SQL syntax.
- 12.Press the F3 key to build the query. The completed query will be displayed.

AD089050

This PTF only applies to OnDemand Distribution Facility installations utilizing an OnDemand V7 archive.

This PTF adds the SQL Tokenized query capability to ODF. ODF has been modified to generate tokenized queries when editing or adding a distribution that will require a report segment selection using an SQL query. This feature will improve query performance on the OnDemand Server by using the DB2 prepare cache. Once installed, ODF will build and use SQL queries in the tokenized format. However, existing queries will not automatically be converted to the tokenized format. A new transaction has been added to the ODF Batch Utility to provide for selective or mass conversion of all the queries that are currently in the ARSPQT table. The transaction is only valid on the SYSIN transaction definitions.

To selectively change a distribution's queries, you can specify a single Bundle Entry as follows:

```
REB BQT  
K BQT_DIST_ID=A00001  
K BQT_DIST_NAME=DIST A REPORTS  
K BQT_SEQUENCE=10
```

To change an entire distribution's queries, you can specify a single distribution using a range specification as follows:

```
REB BQT  
K BQT_DIST_ID=A00001  
K BQT_DIST_NAME=DIST A REPORTS  
N BQT_DIST_ID=A00001  
N BQT_DIST_NAME=DIST A REPORTS
```

To change a range of distribution queries, you can specify the advantage of the DB2 prepare Cache. Once installed, ODF will build and use SQL queries in the tokenized format. However, existing queries will not automatically be converted to the tokenized format. A new transaction has been added to the ODF Batch Utility to provide for selective or mass conversion of all the queries that are currently in the ARSPQT table. The transaction is only valid on the SYSIN transaction definitions.

To selectively change a distribution's queries, you can specify a single Bundle Entry as follows:

```
REB BQT  
K BQT_DIST_ID=A00001  
K BQT_DIST_NAME=DIST A REPORTS  
K BQT_SEQUENCE=10
```

To change an entire distribution's queries, you can specify a single distribution using a range specification as follows:

```
REB BQT  
K BQT_DIST_ID=A00001  
K BQT_DIST_NAME=DIST A REPORTS  
N BQT_DIST_ID=A00001  
N BQT_DIST_NAME=DIST A REPORTS
```

To change a range of distribution queries, you can specify the range of distributions using a range specification as follows:

```
REB BQT  
K BQT_DIST_ID=A00001  
K BQT_DIST_NAME=DIST A REPORTS  
N BQT_DIST_ID=A99999  
N BQT_DIST_NAME=DIST A REPORTS
```

The Batch Utility only reports whether the rebuild of the queries was successfully converted or not. It is possible to rebuild all the queries in the ODF database.

However, if the ODF database is large, we recommend that you specify smaller changes of distributions to be converted. The batch utility will only perform a commit after the rebuild transaction completes, so the large range could be on a very large span of control for the commit. ODF supports both formats of the query during distribution processing, so it is not necessary to convert the queries on installation of this PFT.

Note: The new format will generate a larger query that is stored in the ARSPQT tables query field. The maximum size of a query is 32000 bytes. If a query will exceed the maximum length, the non-tokenized query will be built.

AD12309O

This PTF only applies to OnDemand Distribution Facility installations utilizing an OnDemand V7 archive. This PTF provides a performance enhancement when distributions have multiple bundle entries defined. The principle change was to maintain the connection to the server for the life of the distribution and to provide direct access to the OnDemand Archive database. This eliminates the open and close that is done during normal distribution processing and improves the database access by eliminating the call to the OnDemand ARSDOC command. ODF distributions that have been defined with only a single bundle entry may not see much improvement in performance with this PTF. This feature must be enabled in order to be used by ODF. To enable, clear the DD DUMMY JCL statement //ARSNODOC DD DUMMY by removing the Asterisk (*) in column 3.

AD16618O

This PTF only applies to OnDemand Distribution Facility installations utilizing an OnDemand V7 archive. This PTF provides a performance enhancement for distributions that have multiple bundle entries defined and query the archive server for document selection. This feature will perform a check of the documents existence on the OnDemand Archive server during distribution processing. When the document does not exist, a print request for that document will not be inserted in the database. This saves the processing done during printing that handles the not found condition. This can be a significant reduction in CPU usage and the elapsed time to print. Before enabling this new feature, be aware of the following:

1. Document not found messages will no longer appear in the messages or manifest output.
2. Large distributions with many bundle entries that query the database for a document will take much longer to schedule for printing.
3. The distribution processor that releases a distribution for printing is single threaded. A large number of requests that arrive at the same time may causes delays in printing.

This feature must be enabled in order to be used by ODF. To enable it, edit the ARSODFC1 member found in the SARSJCLS library and change the VERIFYQUERY value from an N to a Y. A value of M can be used; this value will apply this feature to the Continuation Processor only, and the Main distribution processor will not use this feature. This feature will only work with OnDemand V7 Archive; if the ODF system defines a V2 or V2 and V7 system, this feature will be disabled. The setting for the server is found in the SDT 23 definition; the first four characters must be defined as 'V7' to use this feature.

A new batch facility is also being provided with this PTF. This batch facility will perform the same but limited processing done in the real-time distribution processor in batch mode. This process will only insert print requests for documents that exist on a V7 Archive Server. This batch process will perform only on one specific distribution at a time. The batch distribution processor is designed to work with an external process that inserts the Document Status records (DSTs) for this batch facility to process. The external process would set the status field to a value of 'Q'. The real-time distribution processor will ignore this status code so that the batch distribution processor can act on it.

A JCL member has been provided to be modified to conform to the installation standards and incorporated into user provided job streams. The member name is ARSBDSTJ, and is found in the SARSJCL5 library. The input parameters are as follows:

```
//PARMIN DD *
DB2SSID=DSNA
DB2PLAN=ARSBDIST
REPORT=ON
REPORT=NOHITS
DIST-ID=DISTID
DIST-NAME=DISTRIBUTION NAME
/*
```

Set the DB2SSID and DB2PLAN names to the installation standard names your system and plan uses to execute this program. The REPORT=ON option will write a report line for every PPT inserted for the print processor. REPORT=ON will not eliminate the insert report, but the final totals will still be produced. Set the DIST-ID and DIST-NAME parameters to the distribution that is to be processed by the batch distribution processor. The Batch Distribution processor will only handle one distribution per invocation. You cannot specify more than one set of DIST-ID/DIST-NAME parameters. The REPORT=NOHITS option can be specified to help diagnose why documents are not being retrieved. Additional displays will provide meaningful information about the query requests, but should not be used normally.

How to use Batch Distribution Processor

An external process will create DST records for the selected documents to be processed by the print processor of ODF. This external process is user provided. The DST record must be properly constructed with a valid Report ID, Application Group, Application, and Load ID for a defined distribution with the DST_STATUS field set to a value of 'Q'. The Batch Distribution processor would be invoked with the Distribution ID and Distribution Name specified on the PARMIN dd statement. When invoked, the Batch Distribution Processor will validate the distribution, then retrieve the DST records, and find the matching bundle entries defined for the

distribution. If the bundle is defined with a query selection, the query will be retrieved and passed to the OnDemand server to determine if the document exists for the query. Only documents that exist will have a print request inserted (PPT). If the bundle entry defines a full document print, a PPT will always be inserted. Once all DST entries have been processed for every bundle entry, a DRT record will be inserted to indicate to ODF that the distribution is ready for printing. When ODF detects the DRT status Q record, it will invoke the print processor immediately. No further bundling will take place. This feature assumes that all the bundling for the distribution has been done, so it does not recognize the Wait/Ignore indicator that is set in the bundle entry.

AD18908O

This PTF only applies to OnDemand Distribution Facility installations utilizing an OnDemand V7 archive. This PTF provides a performance enhancement by running ODF with a local copy of the ARSSOCKD server, which eliminates the use of the TCP/IP communications protocol. The improved throughput will vary by the type of distributions defined to ODF. Large distributions with many bundle entries that perform segment selection will see the greatest improvement. This feature can only be used when processing distributions using subtasks.

Submitted jobs is not supported. This means that distributions defined with a Job Name on the Distribution Control Table entry (DCT) and the Bundle Definition Table entry (BDT) will not be able to take advantage of this feature.

AD30814O

This PTF applies to OnDemand Distribution Facility installations utilizing an OnDemand V7 archive. This PTF provides a performance enhancement for distributions that have Job Name controls established by Report Administration. Address Spaces dynamically created for Job Name controlled distribution processes will remain persistent for as long as there is distribution work to be handled. Contrast this to a process that initiated fresh controls for each new distribution. Persistent distribution processing will capitalize on economies gained through reusing resources.

16.2 Report Distribution

OnDemand Report Distribution provides an easy way to automatically group reports and portions of related reports together, organize them, convert the report data into different formats, and send them through e-mail to multiple users or make them available for printing.

16.2.1 Report Distribution components

Report Distribution consists of the following components:

- ▶ Report

A report in Report Distribution is a document or a set of documents that are retrieved from the OnDemand system to be bundled and delivered to one or more users. A report can be e-mailed to the users or sent to their default server printers. Before retrieving a report, the documents must have been loaded into OnDemand. Documents can be loaded in the following formats:

- AFP.
- Line data.
- Unformulated ASCII data.
- PDF.

Reports can be retrieved using the following methods:

- Load: Building a list of documents based on the documents that are loaded during a specific time frame. This method is associated with application groups.
- Named Query: Performing a database query using a public named query that was defined by the OnDemand Windows Client. This method is associated with folders.
- SQL: Performing a database query using an SQL query. This method is associated with application groups.

- ▶ Banner

A banner is a page that is printed at the start of, within, or at the end of a print job. The banner uniquely identifies the output. A banner can contain information about the distribution, its contents, the bundle, the reports, and the recipient that received the distribution. The recipient information is taken from the user information that was defined when the user was created.

Banners are optional in a report distribution. If you choose to use banners in a distribution, you must add them to a bundle, and add that bundle to a distribution. You can choose to use three different types of banners:

- Header banner: Placed before all the reports in a bundle.
- Separator banner: Precedes each report in the distribution.
- Trailer banner: Follows all of the reports in a bundle, and is placed before the manifest. If the manifest is included in the distribution, the three types of banners can contain different information from the distribution and the recipient user IDs of the distribution.

▶ **Bundle**

A bundle is an OnDemand Report Distribution object that allows you to package, organize, and optionally provide additional information about the reports that you want to send to the recipients. A bundle contains at least one report, and can optionally include banners and a manifest.

A distribution contains a single bundle; however, a bundle can belong to more than one distribution.

▶ **Schedule**

A schedule determines when and how often OnDemand sends out a distribution. A schedule can be time-based or load-based. OnDemand allows you to set the distribution once, daily, weekly, or monthly. If you set the schedule to be load-based, OnDemand sends out the distribution as documents that are required for the distribution are loaded into the system.

▶ **Recipient**

A recipient is an OnDemand user or group that is assigned to receive reports using report distribution. When you create a distribution, you assign who should receive the reports that are contained in the bundle. If a user or a group of users are recipients of a distribution, they can receive and view all of the reports in that distribution even if they do not have permission to view these reports from the OnDemand Windows Client. For example, a group of users do not have permission to view a customer's purchase orders from the OnDemand archive in general; however, if they are in the recipient list of a distribution that includes these orders, they still receive them even though they may not have permission to view these reports from the OnDemand Windows Client. A recipient list contains all of the recipients of a distribution. A recipient list can contain a combination of individual users and groups. Recipient lists are associated with distributions only, not with any other object in OnDemand.

▶ **Distribution**

A distribution consists of a set of reports that are contained in a bundle, one or more recipients to receive the reports, and a schedule that specifies when the distribution is delivered. All of the recipients receive all of the same reports in the same format. You use the OnDemand Administrator to define characteristics of a distribution in the OnDemand system. A distribution includes a distribution name, a bundle, one or more recipients, and, optionally, schedules' notes.

16.2.2 Setting up Report Distribution

To set up deliveries of your distributions, you need to complete the following tasks:

1. Define reports.
2. Define banners (if you want to use them).
3. Define bundles of reports that are used by distributions.
4. Define distributions.
5. Identify the distribution schedule.
6. Add recipients of distributions.

Defining a report

To define a new report:

1. Start OnDemand Administration Client.
2. Expand **Report Distribution**.

3. Right-click **Reports** and select **New Report** from the pop-up menu. See Figure 16-15.

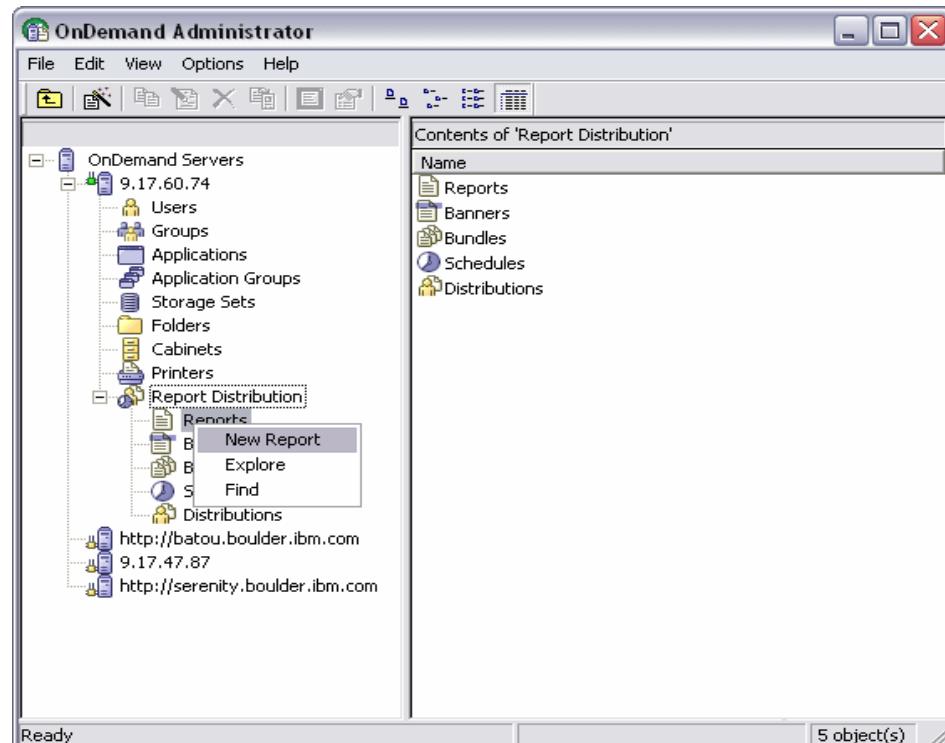


Figure 16-15 Defining a report

4. Type the name of the report you want to add.

5. Select the report definition type (Load, Named Query, or SQL) and select the associated fields. See Figure 16-16.

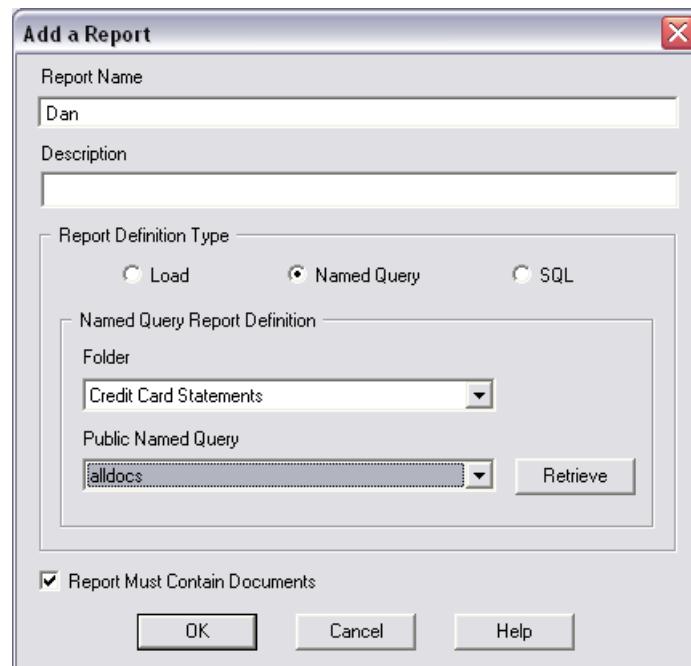


Figure 16-16 Adding a report

6. Click **OK**.

Defining a banner

To define a banner:

1. Right-click **Banner** and select **New Banner** from the pop-up menu.
2. Specify the banner name and the banner type you want to use and the header banner information. See Figure 16-17.
3. Click **OK** to save the banner information.

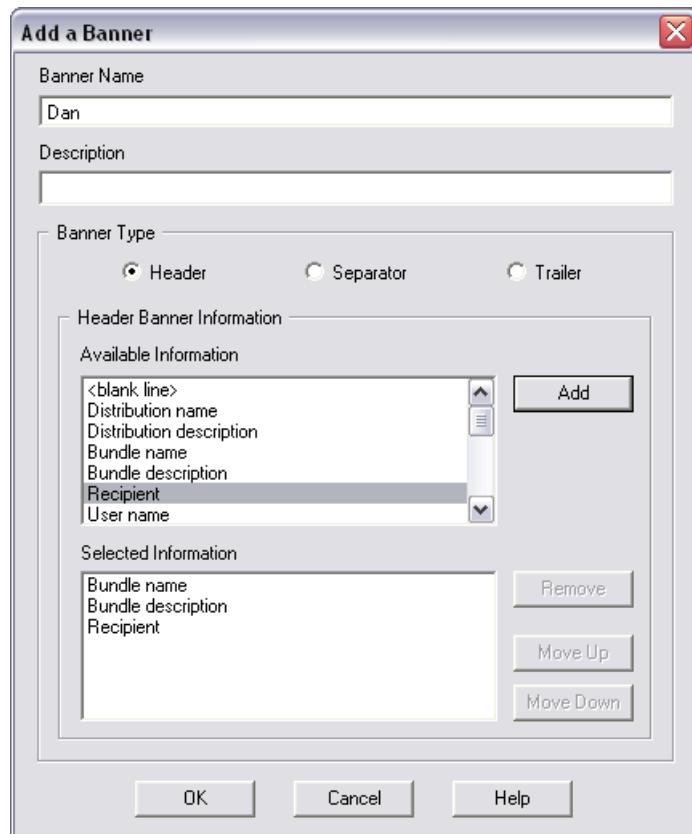


Figure 16-17 Adding a banner

Defining a bundle

To define a new bundle name:

1. Right-click **Bundles** and select **New Bundle** from the pop-up menu.

2. The Add a Bundle window displays, as shown in Figure 16-18.

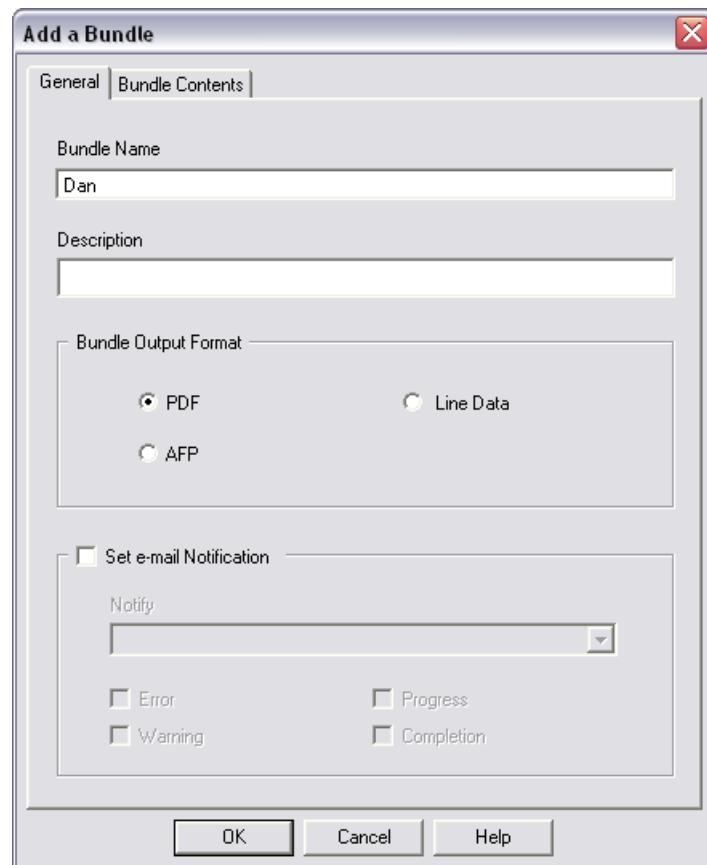


Figure 16-18 Creating a bundle

3. Under the General tab, type the bundle name and the banner type you wish to use. See Figure 16-19.

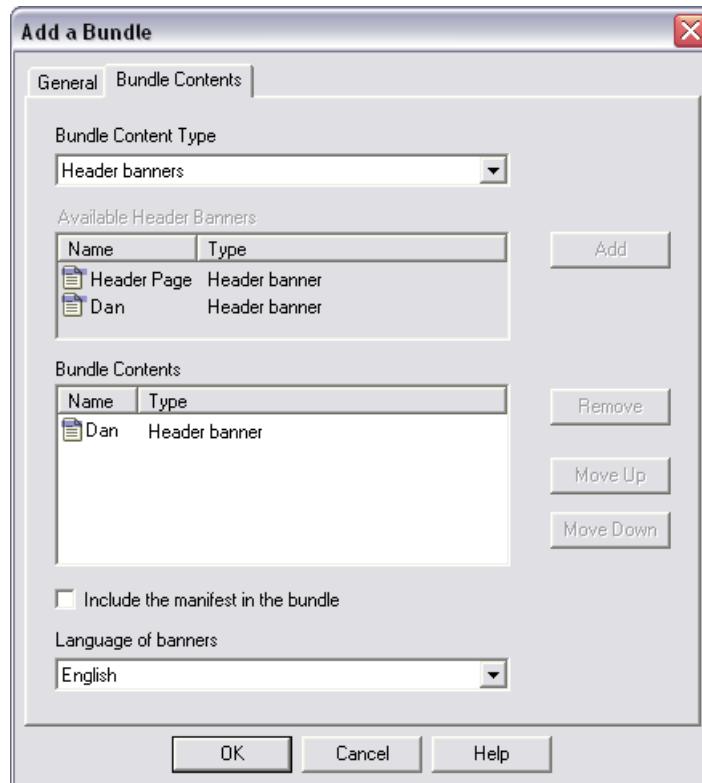


Figure 16-19 Defining a bundle

4. Define the bundle contents you are going to use. More than one content type can be added. For example, you could combine header banners and reports.

To add both a banner and a report, select the content you want to bundle under bundle content type and select the **Add Tab** button so it appears under the bundle contents. See Figure 16-18 on page 561.

5. Click **OK**.

Note: Your system *must* be set up to use a third-party transform (AFP2PDF or Xenos d2e transform) if the data types in the input files are different, for example, PDF and AFP. The transform parameters are configurable in the ARS.CFG file.

Defining a schedule

To define a schedule:

1. Right-click **Schedule** and select **New Schedule** from the pop-up menu. See Figure 16-20.

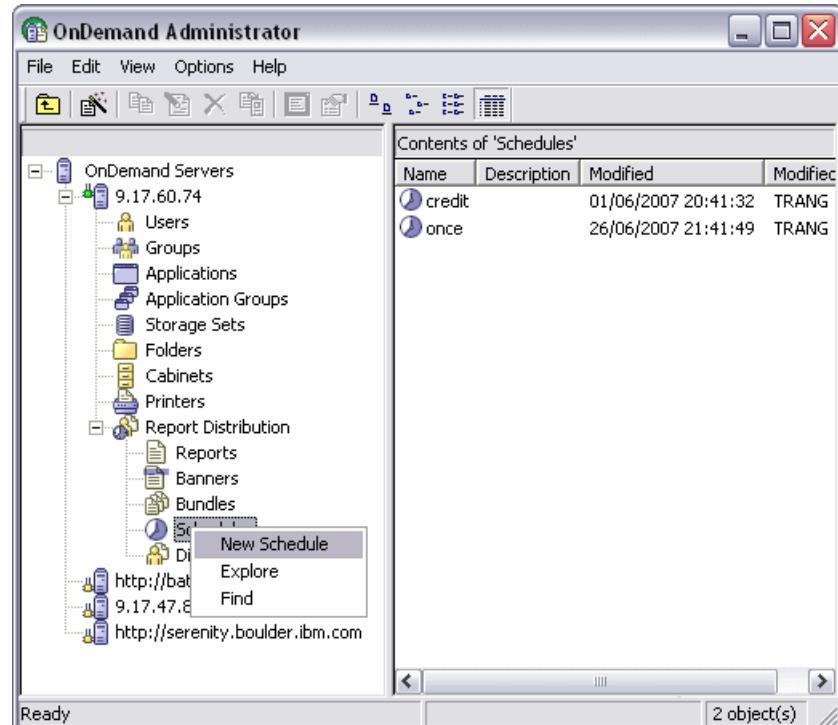


Figure 16-20 Defining a schedule

2. Define a unique name for your schedule and select how often you want the schedule to run. See Figure 16-21.
3. Click **OK** to save your changes.

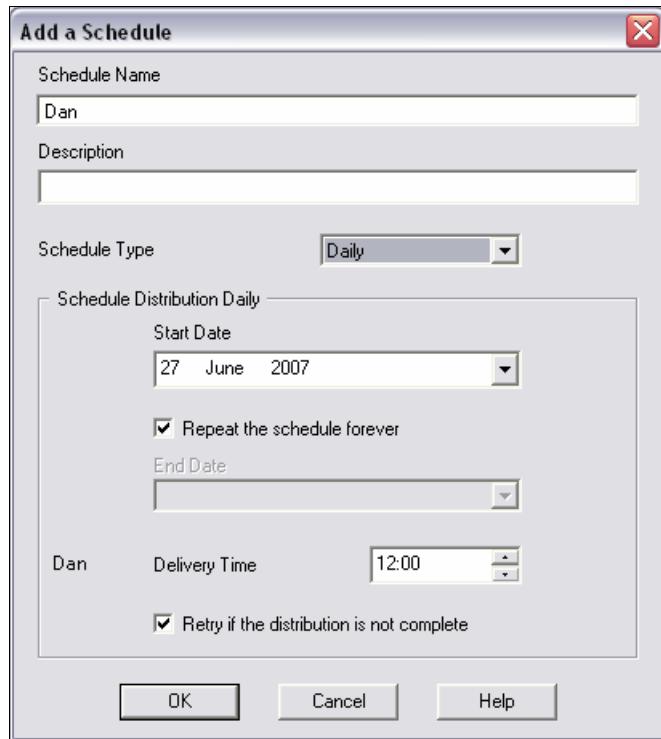


Figure 16-21 Adding a schedule

Defining a distribution

To define a new distribution name:

1. Right-click **Distribution** and select **New Distribution**, as shown in Figure 16-22.

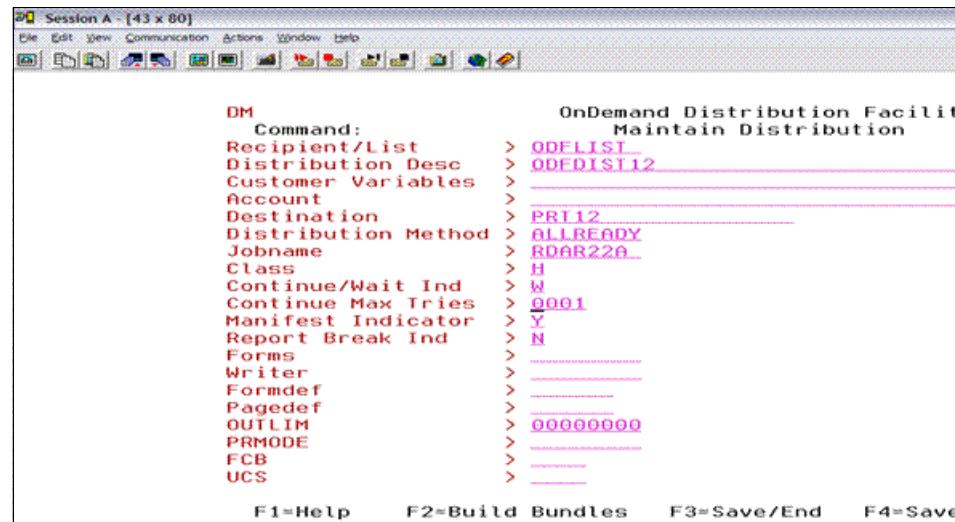


Figure 16-22 Defining a distribution

2. Click the **General** tab. Define a unique name for your distribution and select the delivery options for the distribution. If applicable, select who to notify. See the sample setup in Figure 16-23.

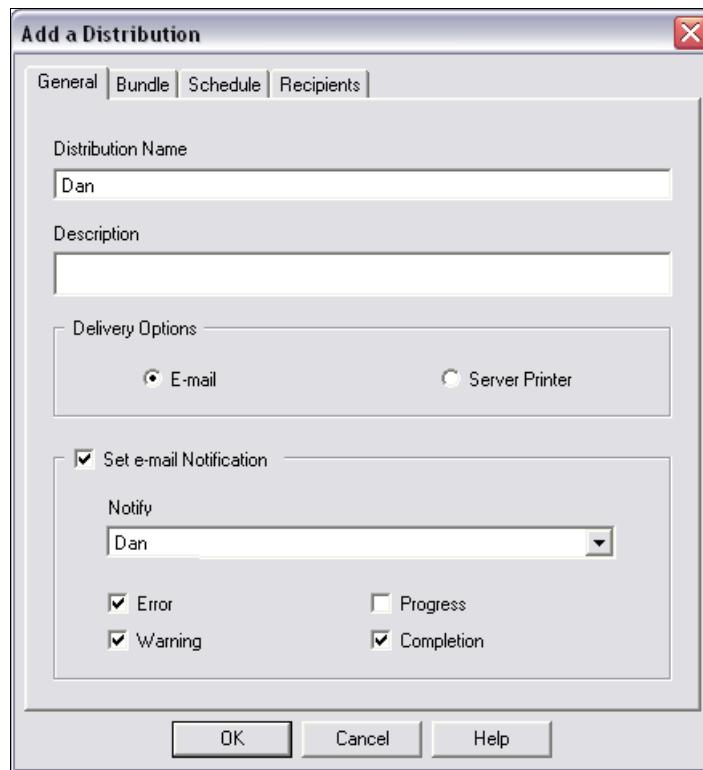


Figure 16-23 Defining a distribution name

Tip: The Server Printer delivery option is specific to InfoPrint and does not support a local or network defined printer.

3. Click the **Bundle** tab. Select the distribution bundle you are going to use. See Figure 16-24.

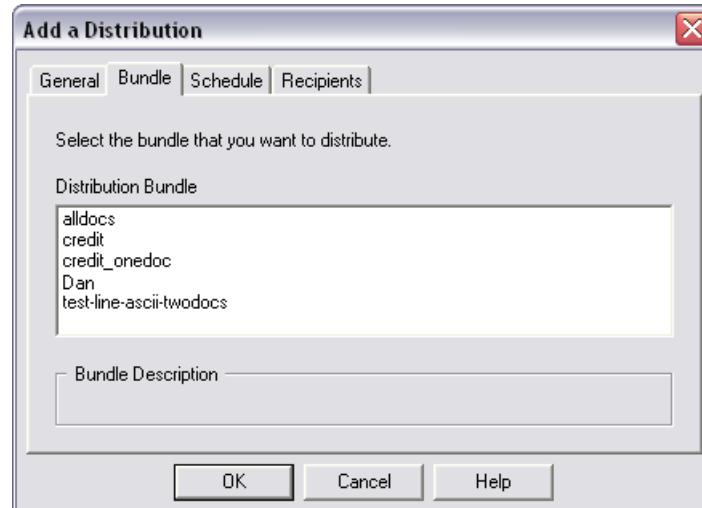


Figure 16-24 Defining a distribution bundle

4. Click the **Schedule** tab. Select the distribution schedule you are going to use. See Figure 16-25.

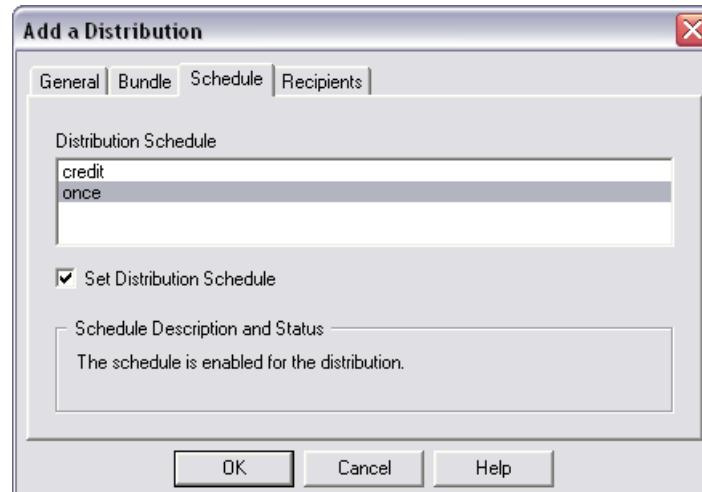


Figure 16-25 Defining a distribution schedule

5. Click the **Recipients** tab. Select the recipients you wish to add by highlighting the recipients name and clicking the **Add >>** button. See Figure 16-26 on page 568.
6. When complete, click **OK** to save the information.

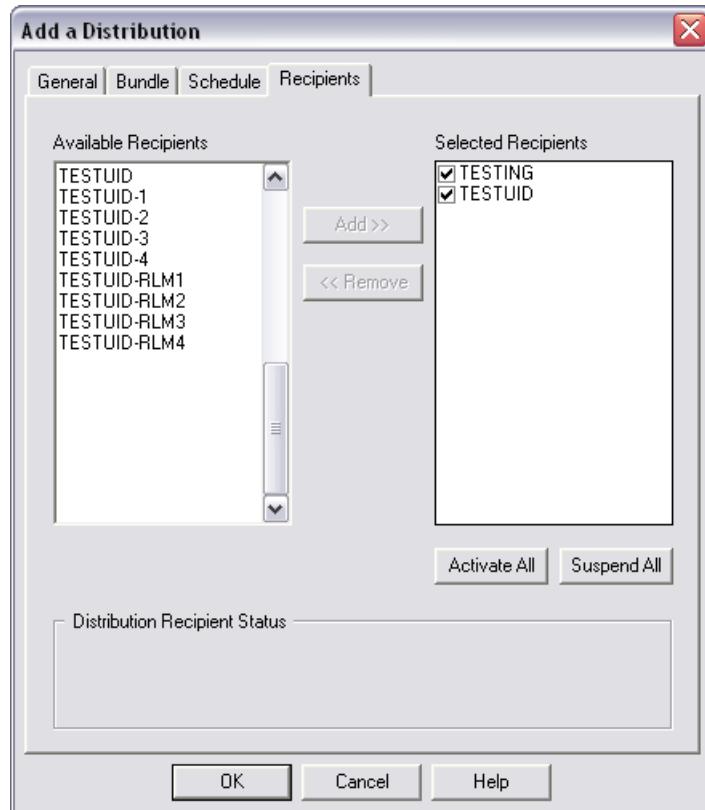


Figure 16-26 Adding a recipient

16.3 Content Manager OnDemand Toolbox

The Content Manager OnDemand Toolbox was developed by the Content Manager OnDemand Support for the IBM System i™ group and it runs on all platforms.

The OnDemand Toolbox is available as compiled binaries. It is written in Visual Basic and released as open source software under the IBM Public License.

It was initially planned as a package of open source code examples that could serve as a base for custom software development. This was later extended to a complete toolbox, containing complete applications that can be used by the customer or by the partner to:

- ▶ Modify and update key-fields in existing documents
- ▶ Delete documents
- ▶ Manually store new documents into Content Manager OnDemand
- ▶ Automatically archive files found in a directory into Content Manager OnDemand

The OnDemand Toolbox allows partners and customers to customize the application to suit their needs.

16.3.1 OnDemand Toolbox installation

Installation is extremely simple. Follow these steps:

1. Download the OnDemand Toolbox from the IBM Redbooks Web site (see Appendix A, “Additional material” on page 605).
2. Run setup.exe.

The installer will guide you through the setup process.

16.3.2 OnDemand Toolbox components

The OnDemand Toolbox consists of four separate applications: OD Store, OD Update, OD Delete, and OD Filesystem Monitor. Each application is a single tool that is designed to provide exactly one key feature of the toolbox. Each component of the toolbox is independent. This enables administrators to deploy only specific parts to the user’s computer. The only base is a package of some common libraries that are installed with all of the components.

OD Store

OD Store is a simple application that allows users to manually archive documents from their PC directly into the Content Manager OnDemand.

The user selects one or more files and enters the key fields by hand.

Documents can only be stored into an existing application, application group, and folder. Also, an administrator has to define which type of files (for example, Word files are *.doc files or Excel files are *.xls files) users are allowed to store into which storage-set within the Content Manager OnDemand.

OD Update

Using this tool, a user can search for documents and view documents the way the standard OnDemand Windows Client works.

In addition to this task, the user can select one or more documents and modify their key fields.

OD Delete

OD Delete has the same interface as OD Update. It allows the user to search for documents and review them. Instead of update functionality, it allows the user to delete one or more documents.

OD Filesystem Monitor

The OD Filesystem Monitor is a server program that runs without any user interaction.

It serves as a directory monitor. It monitors one or more directories and archives the files found in that directory.

The OD Filesystem Monitor needs to be configured by an administrator before it can be used. For configuration, it needs information about which file type shall be archived using which internal storage set (application, application group, and folder). The metadata for the archived document will be extracted from the file's metadata. The file name, file size, directory name, and some other information can be used here.

16.4 E-mail Notification and Delivery for Multiplatforms

Content Manager OnDemand for Multiplatforms e-Mail Notification and Delivery is a services offering. The services offering provides the following options for the customer:

- ▶ Notify a user through e-mail that an individual document has been loaded into OnDemand.
- ▶ Send that specific document as an attachment to e-mail.
- ▶ Optionally, invoke a transform from AFP or Metacode to PDF, or AFP to DHTML.
- ▶ Include custom text in the e-mail for each user receiving an e-mail notification.
- ▶ Include a link to the OnDemand logon Web Site in the e-mail.

The e-mail is generated and delivered from the OnDemand server in one of two ways:

- ▶ When a document is loaded into the system. This option is primarily for bulk e-mail. You can define profiles to determine which documents generate an e-mail.
- ▶ From the OnDemand Web client (ODWEK) or the Windows Client, by using the FAX interface. Instead of sending the document to a FAX device, the user can send the document to an e-mail address.

To use the transform option, the appropriate transform software, such as Xenos or AFP2WEB, must be used. To use the Metacode to PDF transform option, the customer Xenos transform must be used and the system should be running OnDemand Version 7.1 or later.

OnDemand e-Mail Notification & Delivery is supported on the following platforms: AIX, HP-UX, Solaris, Windows NT, and Windows 2000 Server.

16.4.1 E-mail Notification and Delivery sample output

Below is a example using the **arsmail ins** command to populate a database:

```
arsmail ins -e testuser@us.ibm.com -a A -f "CRD" -g "REMITTM1" -i  
"Account - '000-000-004" -n 'Test User' -x PDF -d afp -t custom_msg.txt  
-s "Your July Statement."
```

The database contains the following row information:

Recipient:	Test user
Email Address:	testuser@us.ibm.com
Transform Type:	PDF
Delivery Type:	A
Application Group:	REMITTM1
Folde:	CRD
Where Clause:	Account = '000-000-004'
Daa Type:	afp
Reply To:	
Subject:	Your July statement

Issue the following **arsmail qry** command to show content of the database:

```
arsmail qry Recipient | Email | Addr | Transform | Delivery | App Group  
| Folder |
```

The command output is as follows:

```
File | Where | Data Type | Reply To | Subject  
Test User1 tuser1@us.ibm.com PDF A CRD CRD test Account= "000-000-004"  
afp  
testuser testuser@us.ibm.com PDF A REMITTM1 CRD customer_msg.txt  
Account = "000-000-004" afp Your July Statement
```

The latest version of E-mail Notification V8.4 is written in Java. The source code is available when the E-mail Notification feature is purchased. You can customize it anyway you want. The other major change is now you can send an e-mail directly to a document. This works as long as the Web interface into OnDemand is WEBI. Upon opening the attachment in an e-mail, the user will be presented with the document the user has been notified about or a login page if the user is not already authenticated. Once authenticated, the document will be displayed.



Enhancements

In this chapter, we explore the enhancements in Content Manager OnDemand.

We cover the following topics:

- ▶ Web Administration Client
- ▶ Composite indexes
- ▶ Cluster indexes
- ▶ Cabinets
- ▶ File name Indexing
- ▶ LDAP security
- ▶ 64-bit support
- ▶ Tracing

17.1 Web Administration Client

OnDemand V8.4 allows you to use a Web administrative client to administer OnDemand. This client enables OnDemand administrators and other individuals who might not be full-time administrators perform certain administrative tasks from the Web browser without having to install the OnDemand Administration Client on their workstations.

The OnDemand Web administrative client allows you to add, view, update, and delete users, groups, applications, application groups, folders, printers, and storage sets.

You can create custom administrative forms by using the Web administrative client and IBM Workplace Forms™ Designer. Individuals who are not full-time administrators can use the custom forms to perform administrative tasks. The Web administrative client also enables users who do not have in depth knowledge of OnDemand to complete administrative tasks.

Launch the Web Administration Client

Launch the Web Administration Client using the following URL (see Figure 17-1):

`http://<host name>:<port number>/ODWebAdmin`



Figure 17-1 Web admin server login window

To log in to a specific server, double-click the server name from the list in the right hand panel, as shown in Figure 17-2.

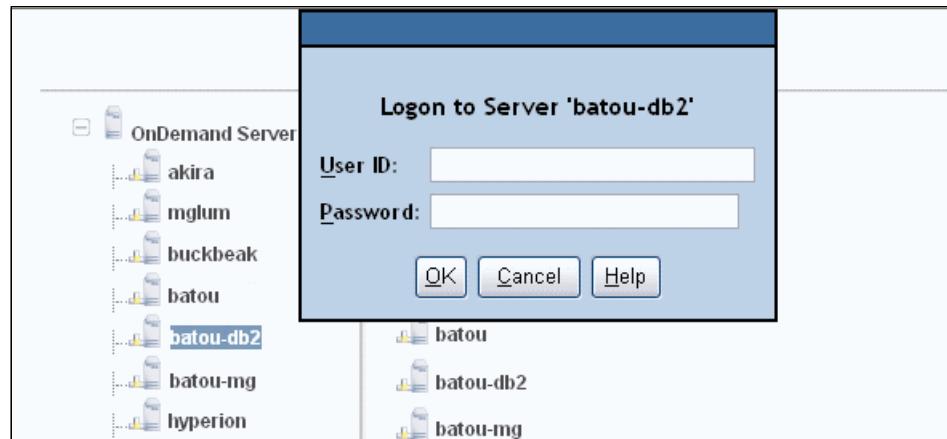


Figure 17-2 Web admin login window

After you complete the login to the designated server, the main Administration Client Windows opens. See Figure 17-3.

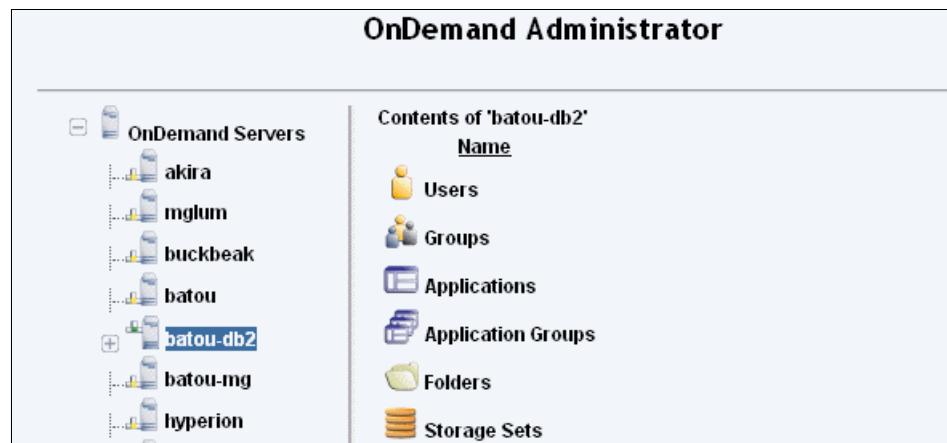


Figure 17-3 Web administration window

View and manage OnDemand settings

The procedures to start adding, viewing, updating, and deleting OnDemand settings (including users, groups, applications, application groups, folders, printers, and storage sets) are similar. We show you the only procedures for viewing and managing users here for your reference.

To view the existing users, right-click **Users** from the right hand side and select **Explore**. See Figure 17-4. You can also expand the OnDemand server on the left hand side, right-click **Users** from the expanded list, and select **Explore**.

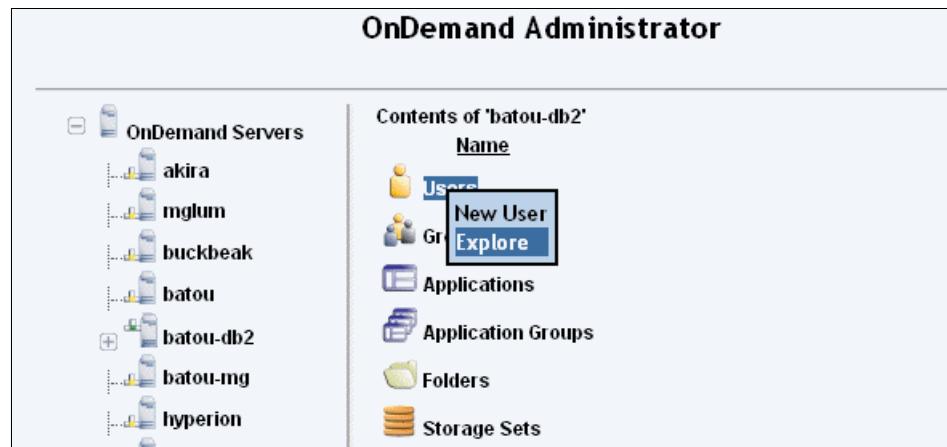


Figure 17-4 Exploring existing users

Figure 17-5 shows a list of existing users.

The screenshot shows the 'OnDemand Administrator' interface. The left pane shows the same tree view of servers. The right pane is titled 'Contents of 'Users''. It displays a table of users:

User ID	Name	Description
ADMIN		OnDemand Administrator
ANEWID		
ANEWUSER	name	desc
ASYS	Debbie Smith	[A] Debbie Smith - Develop
BSMITH	Bob Smith	[A] Bob Smith- Testing
CSMITH	Chris Smith	[A] Chris Smith - Developm
DSMITH	name	desc

Figure 17-5 User list

From the existing user window, you can manage and view a specific user by right-clicking the user and selecting **Copy**, **Update**, **Delete**, or **Properties** from the pop-up menu. See Figure 17-6.

OnDemand Administrator			
		name	desc
	DSMITH		
	SP		
	SP-TEST		
	TESTÁÉÍÓÚÑ		
	TESTING		
	TESTUID	UserID	Used for Regression Test
	TESTUID	Delete	
	TESTUID	Properties	test userid (Leon)

Figure 17-6 Copying user

Use the copy function to quickly create another user with the similar settings.

17.1.1 Requirements and limitations

Web Administration Client uses the `arsxml` utility. It was developed to mimic the Windows administration based client on WorkPlace forms and the Pure Edge Technology. The servlets run under WebSphere Application Server and is the same version as WBI. The WorkPlace plug-in must be installed on the desktop.

Web Administration Client limitations

There are some limitations with using the Web Administration Client. Unlike the OnDemand Windows Administration Client, the Web Administration Client does not come with a Graphical Indexer, Report Wizard, or Report Distribution. It also does not allow you to view and manage cabinets.

Web Administration Client requirements

To use the Web Administration Client, you need to make sure that you are using one of the following supported browsers:

- ▶ Internet Explorer 6 SP1
- ▶ Internet Explorer 7 for Windows XP SP2
- ▶ Firefox 2.0 for Windows XP SP1

At the time of writing, only Windows Server® 2003 R2 is supported for the operating system of the mid-tier application. Windows 2000 or Windows Vista® is not supported.

Always check the support Web site for the latest requirements and additional enhancements for the OnDemand Web Administration Client.

17.2 Composite indexes

A composite index is an index that consists of combined values from two columns in a table. You create a composite index if these two columns are frequently accessed together for a search.

A new tab has been added to the Application Group edit/view window and will only appear if the server is at V8.4.0.0 or higher. Composite Indexes that are created are shown in the Multiple Field Indexes list. Any application group field that has a type of Index will appear in the Single Field Indexes list.

Defining composite indexes is optional. When the application group is viewed, the text on the Update button is changed to Properties. In the database, the index identifier for Multiple Field Indexes is a number (for example, 1, 2, and 3). Rather than using just a number for the index identifier in the dialogs, *Index* is used as part of the identifier and is automatically assigned.

17.2.1 Add a composite index

To add a composite index to an existing application group, follow these steps:

1. From the OnDemand Administration Client, expand the OnDemand server with which you are working.

- Right-click **Application Group** and select **View Application Group**. See Figure 17-7.

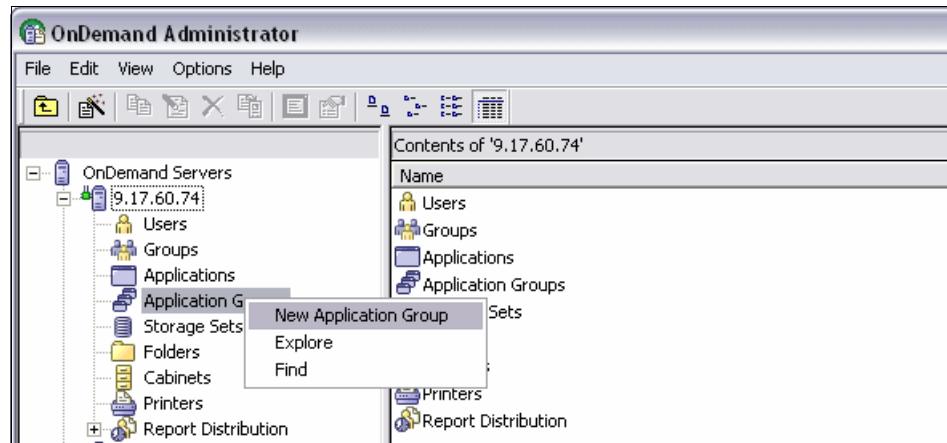


Figure 17-7 Viewing an existing composite index

- Right-click the specific application group and select **Update**. See Figure 17-8.

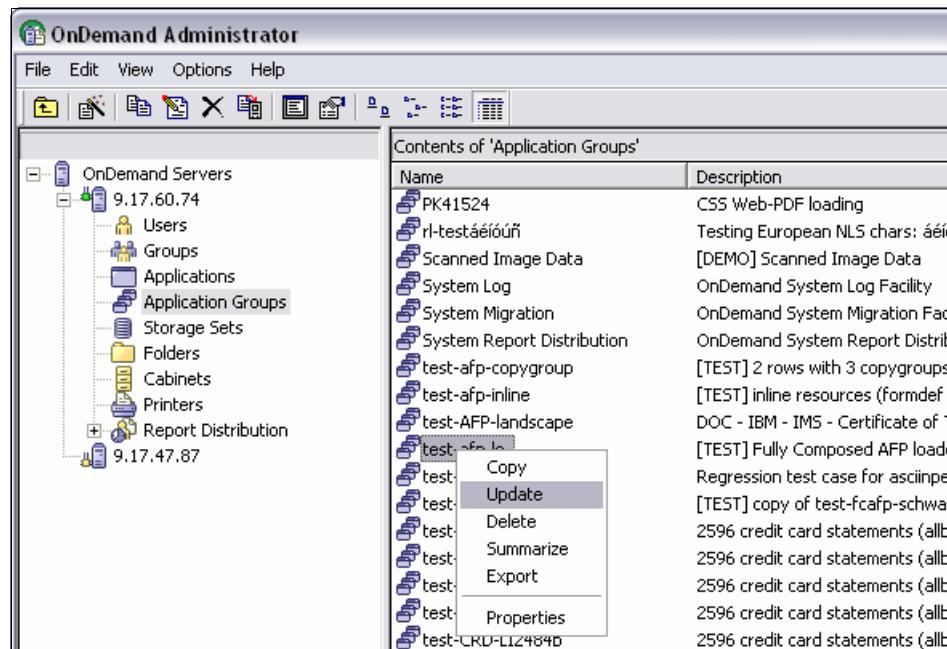


Figure 17-8 Updating an existing index

4. Select the **Advanced Index Information** tab. See Figure 17-9. In the left side pane, it shows the Multiple Field Indexes, if any. In the right side pane, it shows Single Field Indexes.

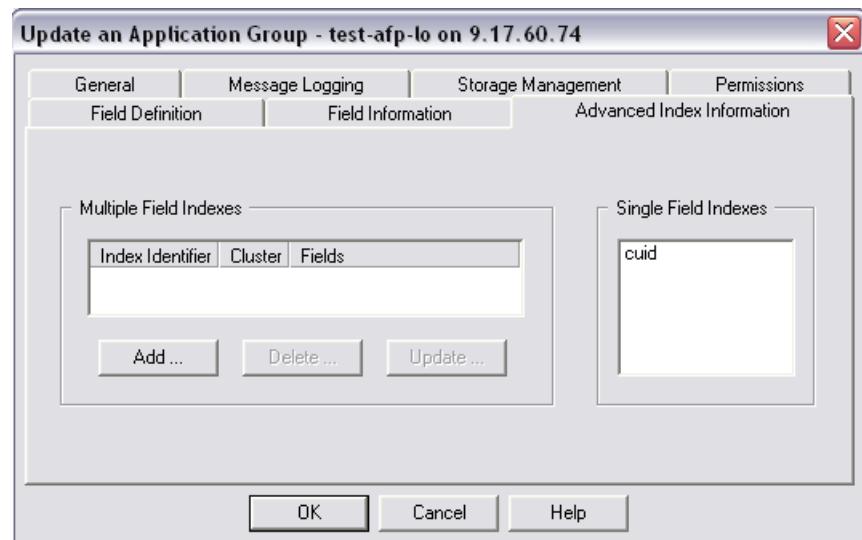


Figure 17-9 Composite index - Update Application Group, Advanced Index tab

5. Click **Add**. This takes you to the Add an Index window.

6. Select at least two indexes from the left Available Fields, and click **Add >>** to add these indexes to the Selected Fields. For our example, we add **cuid** and **report_date** fields as part of the composite index. See Figure 17-10.

Tip: Content Manager OnDemand V8.4 allows up to five indexes to be added.

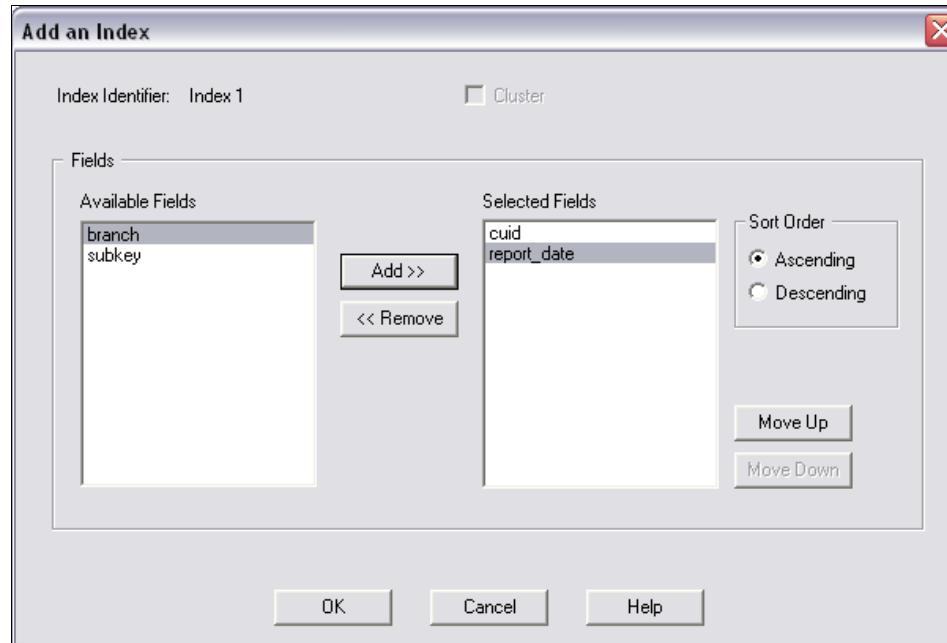


Figure 17-10 Adding multiple indexes

7. Once you add the necessary fields for the composite index, click **OK** to exit the window.

8. Now you should see the composite index you just added under the Multiple Field Indexes pane. See Figure 17-11.

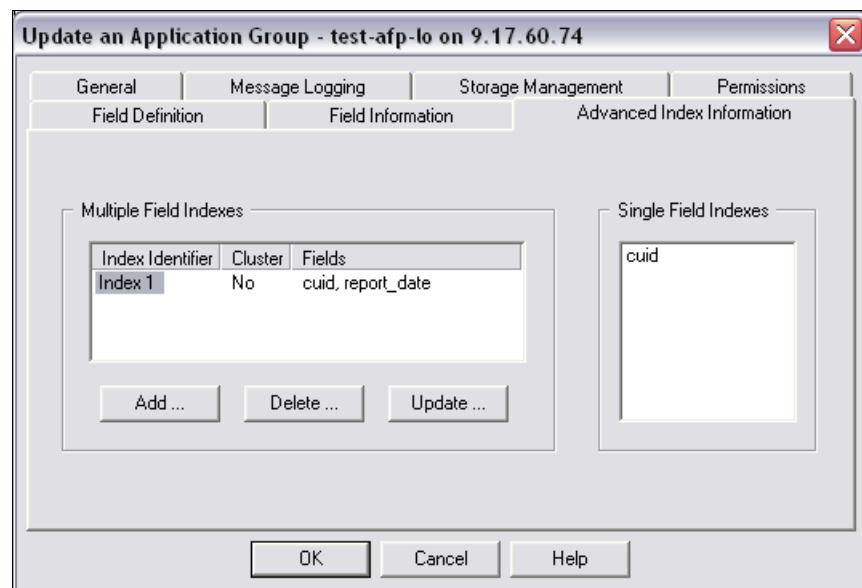


Figure 17-11 Showing added composite index

9. To add another composite index, click **Add** and repeat the previous steps. As an example, we add branch and subkey as the second composite index. See Figure 17-12, which shows both indexes are added.

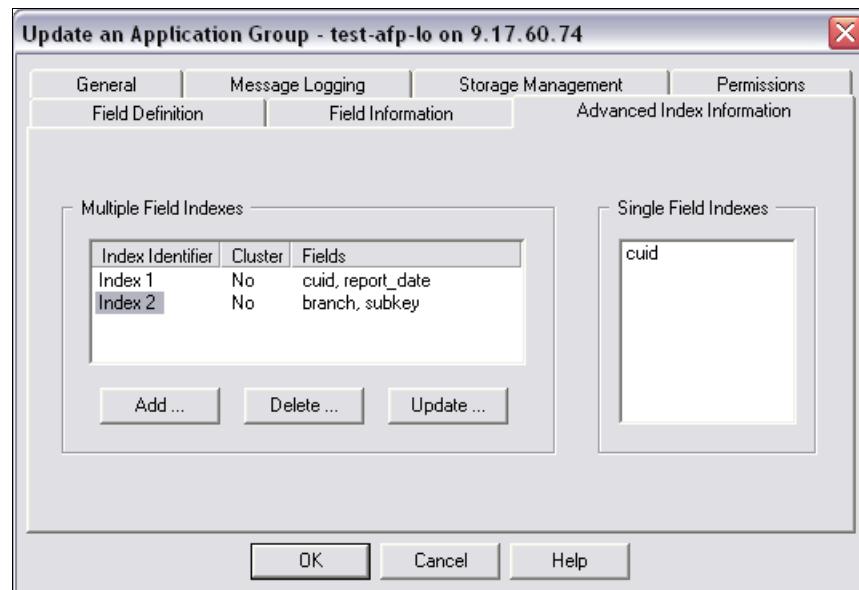


Figure 17-12 Shows indexes that have been added

17.3 Cluster indexes

A DB2 index is a clustering index if the CLUSTER keyword is specified when the index is created. Clustering causes inserted rows to be stored contiguously in sequence whenever possible. Additionally, when the tablespace is reorganized, the data will be sequenced according to the clustering index. Since there can only be one physical sequence for data on disk, there can only be one clustering index per table. If you do not specify a clustering index, DB2 will choose to cluster the data using the oldest existing index. We recommend explicitly specifying a clustering index instead of letting DB2 decide, because you will almost always choose better than the choice DB2 makes.

17.3.1 Define a cluster index

To define a cluster index, we start with creation of a new application group. Follow these steps:

1. From the OnDemand Administration Client, right-click **Application Groups** and select **New Application Group**. See Figure 17-13.

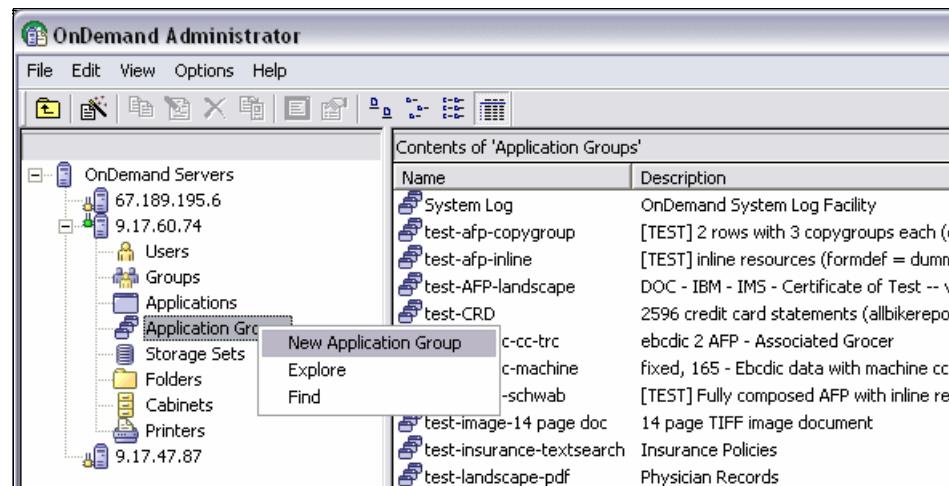


Figure 17-13 creating a new application group

2. Define a name and description for your application group. See Figure 17-14.

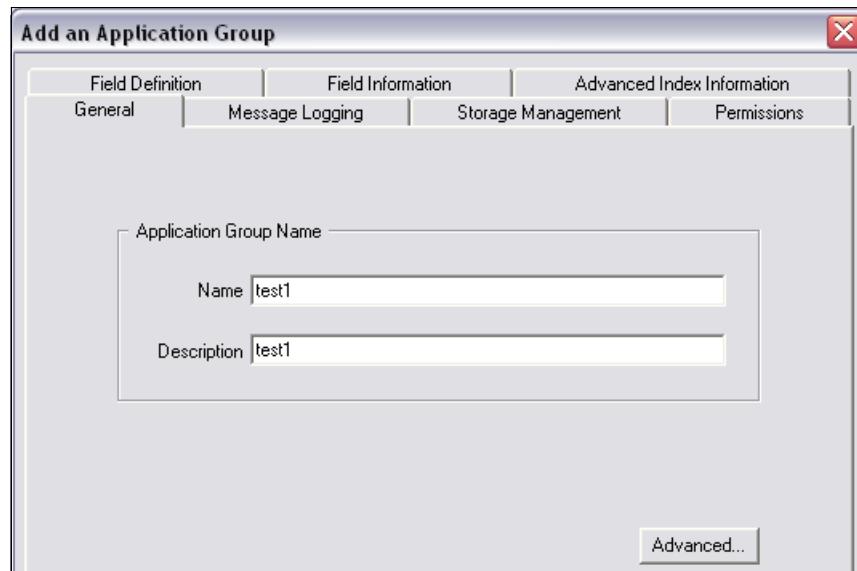


Figure 17-14 Adding a new application group

3. Select the **Storage Management** tab and select a Storage Set Name. See Figure 17-15.

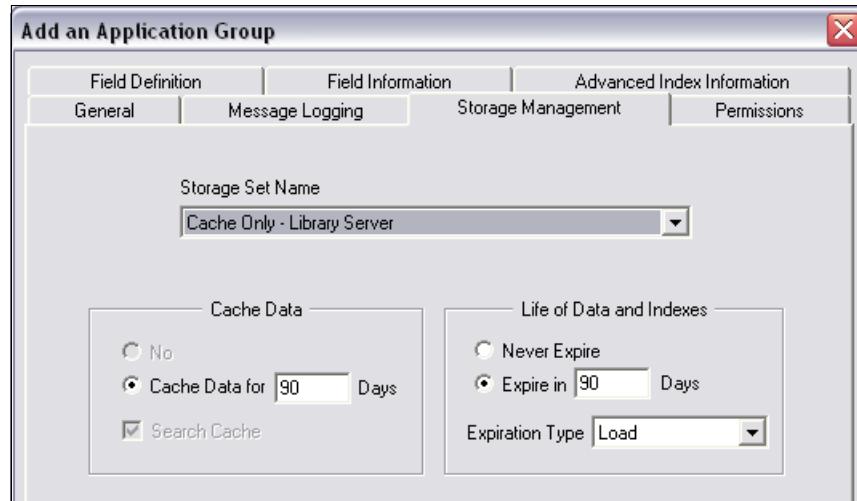


Figure 17-15 Adding a storage name

4. Select the **Field Definition** tab and select a Database Field Name and click **Add**, as shown in Figure 17-16.

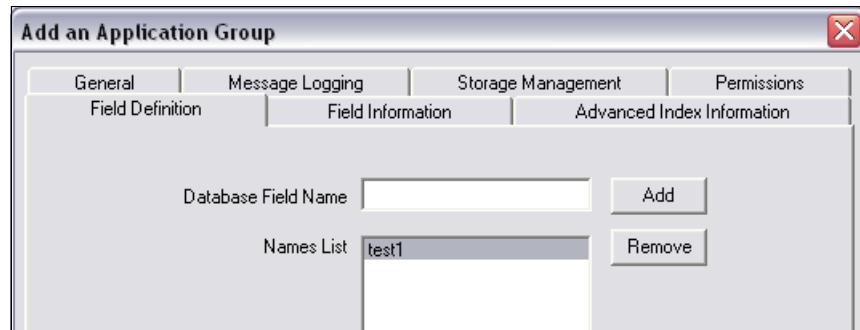


Figure 17-16 Adding a database field name

5. Select the **Field Information** tab and select a string value length for test1. See Figure 17-17.

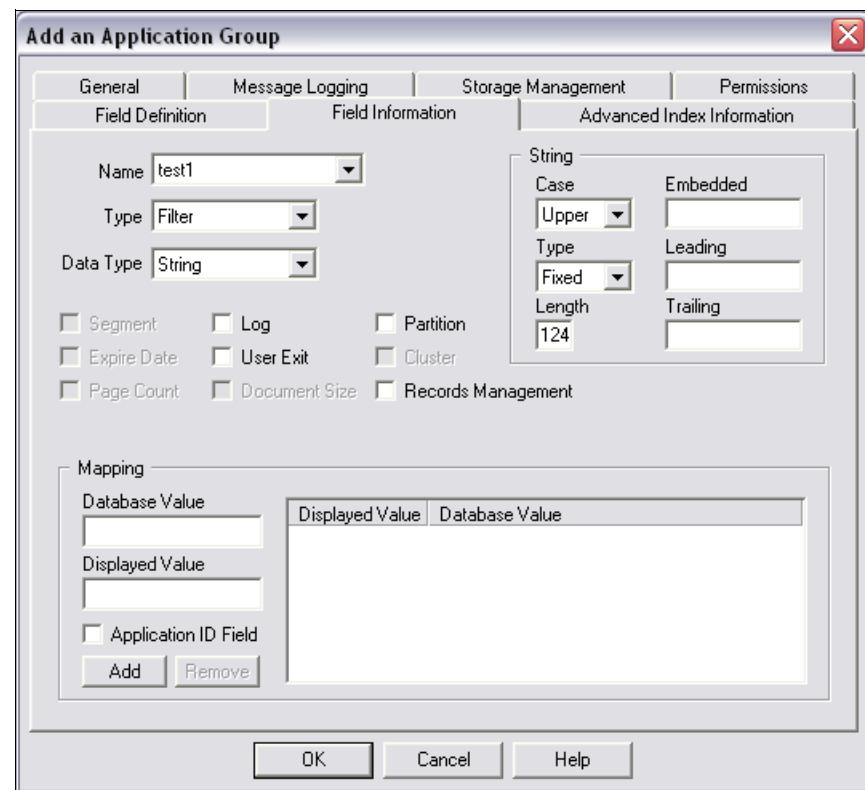


Figure 17-17 Selecting a string value

6. Define a database value and displayed value under Mapping and select **Add** and then **OK**, as shown in Figure 17-18.

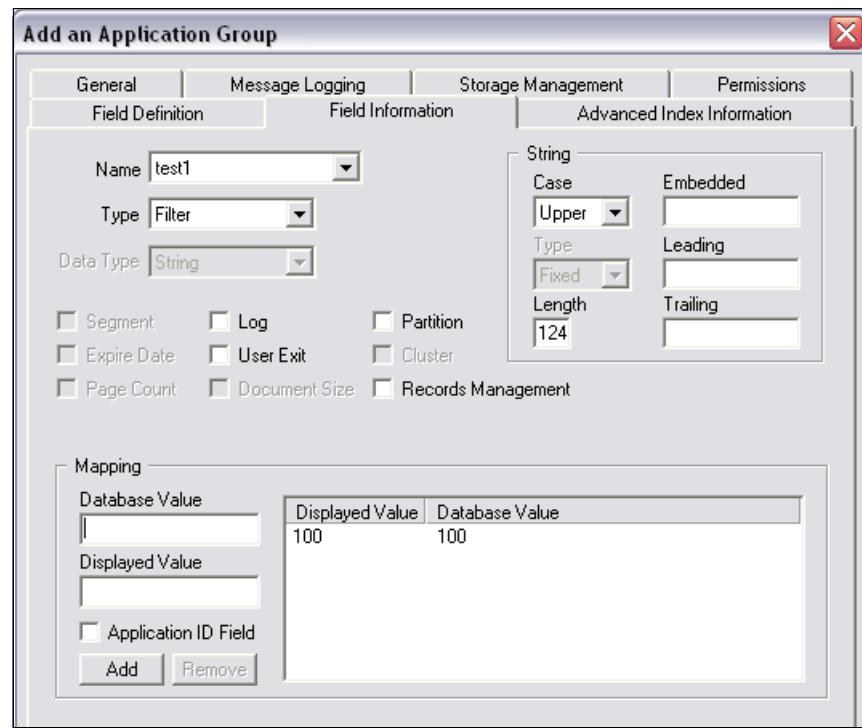


Figure 17-18 Defining a database value

7. Select the **Advanced Information** tab and click **Add**. This will take you to the Add an Index window.

8. Select both fields and click **Add >>** to add them. Select the **Cluster** check box. See Figure 17-19.

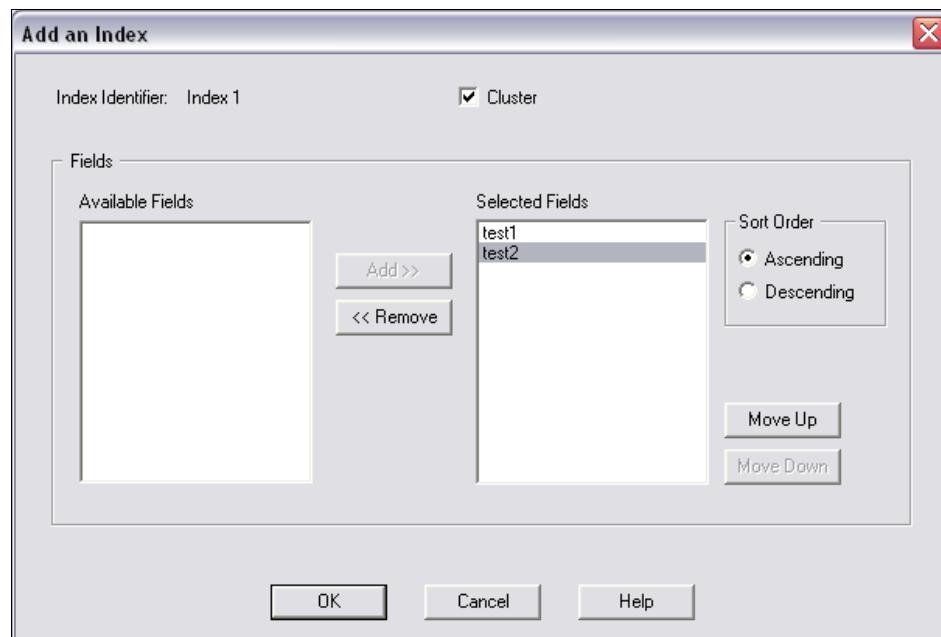


Figure 17-19 Adding a cluster index

9. Click **OK** and this will take you back to the Add an Application Group window. See Figure 17-20.

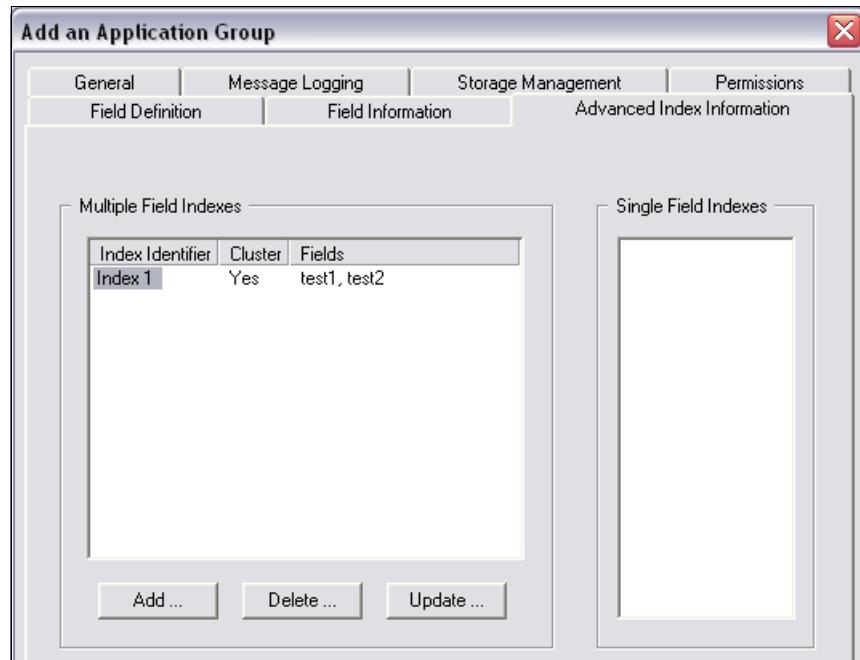


Figure 17-20 Multiple field indexes panel

10. Now you have a cluster index defined within the Multiple Field Indexes. Notice that under the Cluster, Index 1 is set to Yes. Save your application group.

17.4 Cabinets

A cabinet is a container for folders. You can use cabinets to manage folders and enable users to navigate to folders more easily. Figure 17-21 shows the relationship between two cabinets and five folders. Notice, a folder can belong to one or more cabinets.

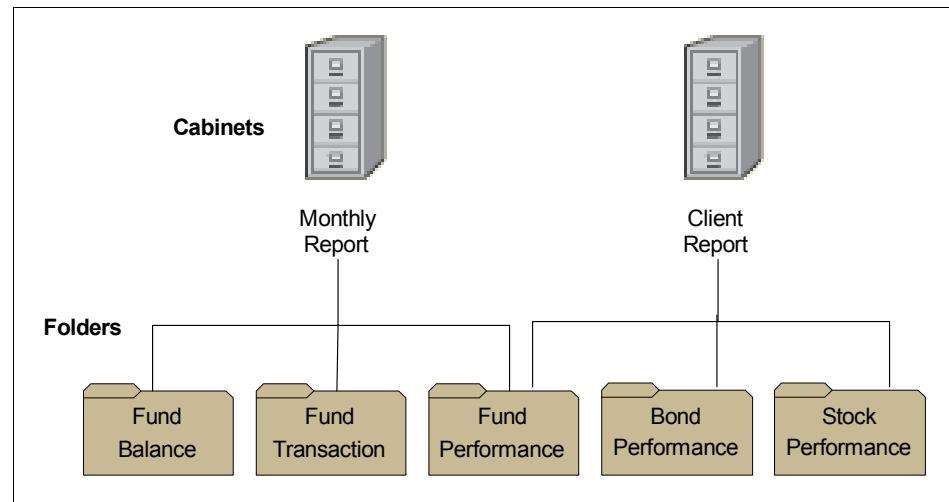


Figure 17-21 Cabinet to folder relationship

17.4.1 Cabinet authorities

You must have Create Cabinets authority or be an application group/folder cabinet administrator or system administrator to work with a cabinet. With the appropriate authority, you can perform add, update, delete, copy, export, summarize, find, explore, or view the properties functions of a cabinet. A user can be given access authority, which enables the user to see the cabinet in a list of cabinets, and view authority, which enables the user to view the contents of the cabinet, as shown in Figure 17-22.

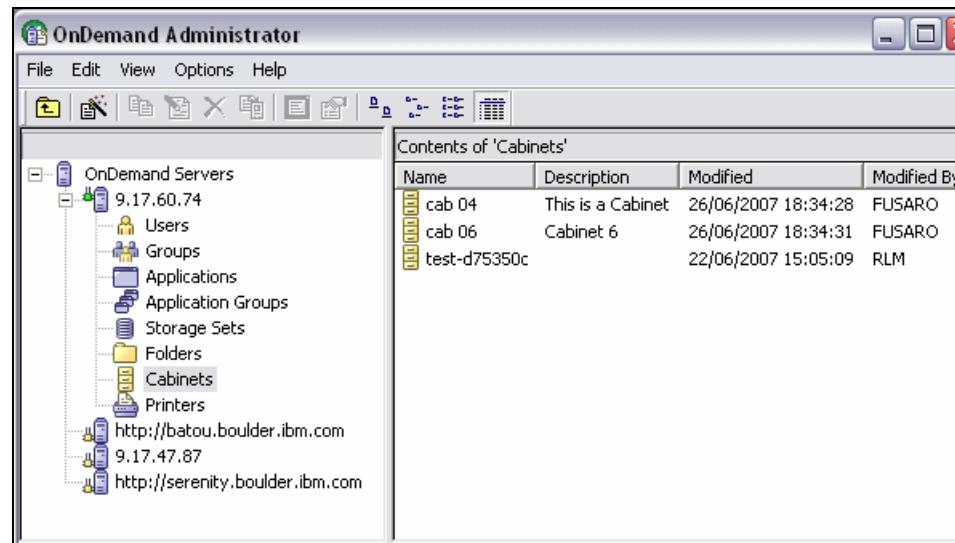


Figure 17-22 Viewing contents of cabinet

You can view the Access and Administrator authority when viewing a specific cabinet and selecting the **Permissions** tab. See Figure 17-23.



Figure 17-23 Cabinet Access and Administrator authority

17.5 File name indexing

File name indexing is new function that has been introduced into Content Manager OnDemand V8.4 to populate an application group database field with the name of the input file at load time.

17.5.1 Set up file name indexing

To set up file name indexing, follow these steps:

1. In the Update an Application Group window, under the Field Definition tab, add the field that will be used by `arsload -b`. For our example, we add `whoDate`. Under the Field Information tab, set the field information. For our example, set `whoDate` type to **Date**. See Figure 17-24.

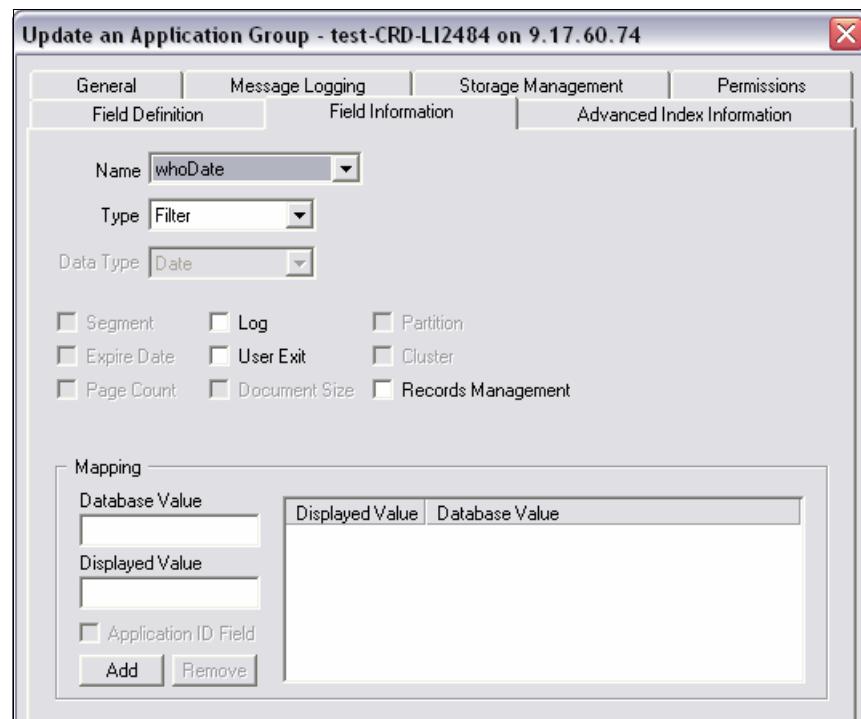


Figure 17-24 Specify format of the date

2. In the Application window, under the Load Information tab, specify the format of the date that will be used in the file name. For our example, we set whoDate format to %m%d%y. This includes date formats such as 102507. See Figure 17-25.

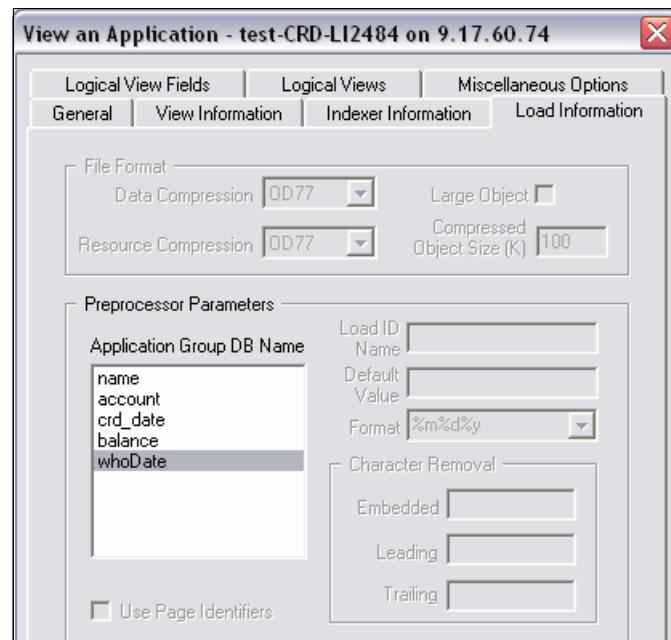


Figure 17-25 Specify the format of the date

3. In the Folder definition, add the field to the folder. Set the field information of the date field appropriately. See Figure 17-26 for the whoDate field set up in the folder.

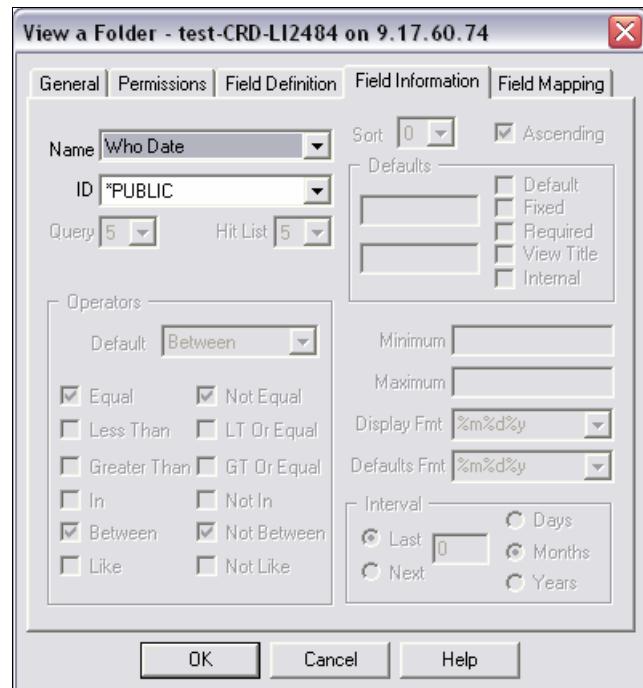


Figure 17-26 Specify the field information for the date field

4. Map the folder fields to the application group fields. See Figure 17-27.

The screenshot shows the 'test-CRD-LI2484 - Search Criteria and Document List' dialog box. The 'Search Criteria' section contains five entries: 'Account' with 'Like' operator and a '%' placeholder; 'Date' with 'Between' operator, values '11/20/93' and '07/19/07', and a 'and' connector; 'Account Name' with 'Equal To' operator; 'Account Balance' with 'Equal To' operator; and 'Who Date' with 'Between' operator and empty placeholder boxes for values and 'and' connector.

Figure 17-27 Map the folder fields to the application group fields

5. Verify that the input file format, depending on the file name format that you specify with the **arsload -b** command. Any of the following input files should work for this test case:
 - MVS.JOB.010199.test-CRD-LI2484.09244.00001.ARD
 - 070907.JOB.DAT.test-CRD-LI2484.07191.1207.ARD
 - MVS.070907.test-CRD-LI2484.test-CRD-LI2484.07191.1207.ARD
6. Running the **arsload** command with the **-b** flag specifies the name of the field/index (application group/application) and the **-B** flag identifies the part of the file name that contains the index value. For example:
 - To process file name 4.a:


```
arsload -d /arsload -u testuid -p TESTPW -b "whoDate" -B
"ign.ign.IDX.ag.ign.ign"
```
 - To process file name 4.b:


```
arsload -d /arsload -u testuid -p TESTPW -b "whoDate" -B
"IDX.ign.ign.ag.ign.ign"
```
 - To process file name 4.c:


```
arsload -d /arsload -u testuid -p TESTPW -b "whoDate" -B
"ign.IDX.app.ag.ign.ign"
```

The results of 5.a (Note the inclusion of the FIELD32/INDEX32 parameters, which were derived from the **-b**, **-B**, and input file name) are shown in Example 17-1.

Example 17-1 Sample arsload output

```
arsload: Processing file
>/arsload/MVS.JOBNAME.010199.test-CRD-LI2484.09244.00001.ARD<
arsload: 07/05/07 14:56:58 -- Indexing started, 458600 bytes to process
0425-415 CC=YES
0425-415 CCTYPE=A
0425-415 CPGID=500
0425-415 FORMDEF=F1ABBB
0425-415 PAGEDEF=P1ABBB
0425-415 INDEXOBJ=GROUP
0425-415 RESTYPE=FDEF,PSEG,OVLY
0425-415 TRIGGER1=*,1,X'F1'
0425-415 TRIGGER2=0,65,X'D7C1C7C5F1'
0425-415 FIELD1=0,2,20
0425-415 FIELD2=1,2,11
0425-415 FIELD3=1,13,8
0425-415 FIELD4=1,71,8
0425-415 INDEX1=X'D5C1D4C5',FIELD1
```

```
0425-415 INDEX2=X'C1C3C3D6E4D5E3',FIELD2
0425-415 INDEX3=X'C3D9C46DC4C1E3C5',FIELD3
0425-415 INDEX4=X'C2C1D3C1D5C3C5',FIELD4
0425-415 USERLIB=/arstest/testcases/data/demo/CREDIT/res
0425-415 FIELD32=X'F0F1F0F1F9F9'
0425-415 INDEX32=X'A68896C481A385', FIELD32
0425-415
inputdd=/arsload/MVS.JOBNAME.010199.test-CRD-LI2484.09244.00001.ARD
0425-415
outputdd=/arstmp/logs/db2/MVS.JOBNAME.010199.test-CRD-LI2484.09244.0000
1.ARD.out
0425-415
indexdd=/arstmp/logs/db2/MVS.JOBNAME.010199.test-CRD-LI2484.09244.0000
.ARD.ind
0425-415
resobjdd=/arstmp/logs/db2/MVS.JOBNAME.010199.test-CRD-LI2484.09244.0000
1.ARD.res
0425-440 ACIF AT PK36252 HAS COMPLETED NORMALLY WITH RETURN CODE 0.
arsload: 07/05/07 14:56:59 Indexing completed
arsload: 07/05/07 14:56:59 -- Loading started, 726008 bytes to process
Resource
/arstmp/logs/db2/MVS.JOBNAME.010199.test-CRD-LI2484.09244.00001.ARD.res
matches the resource >2-1-0<
OnDemand Load Id = >10638-1-0-14FAA-10228-10228<
Loaded 236 rows into the database
Document compression type used - OD77. Bytes Stored = >99993< Rows =
>236<
arsload: 07/05/07 14:57:16 Loading completed
arsload: Processing successful for file
>/arsload/MVS.JOBNAME.010199.test-CRD-LI2484.09244.00001.ARD<
```

7. Figure 17-28 shows a sample document hit list for the document search hit list within the Content Manager OnDemand Windows Client.

test-CRD-LI2484 - Search Criteria and Document List

Search Criteria	
Account	Like <input type="text" value="%"/>
Date	Between <input type="text" value="11/20/93"/> and <input type="text" value="07/19/07"/>
Account Name	Equal To <input type="text"/>
Account Balance	Equal To <input type="text"/>
Who Date	Between <input type="text"/> and <input type="text"/>

Document List						
	Date	Account	Account Name	Account Balance	Who Date	
01/01/98	000-000-000	SMITH CO		935.71	010199	
01/01/98	000-000-000	SMITH CO		935.71	010199	
01/01/98	000-000-000	SMITH CO		935.71	071007	
01/01/98	000-000-001	ADRIAN SHOP		60.14	010199	
01/01/98	000-000-001	ADRIAN SHOP		60.14	010199	
01/01/98	000-000-001	ADRIAN SHOP		60.14	071007	
01/01/98	000-000-002	JACK'S SHOP		1503.67	010199	
01/01/98	000-000-002	JACK'S SHOP		1503.67	010199	
01/01/98	000-000-002	JACK'S SHOP		1503.67	071007	
01/01/98	000-000-003	DENNIS SHOP		88.41	010199	
01/01/98	000-000-003	DENNIS SHOP		88.41	010199	
01/01/98	000-000-003	DENNIS SHOP		88.41	071007	
01/01/98	000-000-004	TERRY'S SHOP		364.93	010199	
01/01/98	000-000-004	TERRY'S SHOP		364.93	010199	
01/01/98	000-000-004	TERRY'S SHOP		364.93	071007	
01/01/98	000-000-005	COBBLESTONE SHOP		2085.75	010199	
01/01/98	000-000-005	COBBLESTONE SHOP		2085.75	010199	

Figure 17-28 Document search hit list

Note:

- ▶ The **arsload** command generates the field/index values slightly differently for the ACIF and PDF indexing parameters:
 - ACIF


```
FIELD32=X'index_value(in hex)'
INDEX32=X'db_field_name(in hex)',FIELD32
```

- PDF

```
FIELD31 = 'index_value'  
INDEX31 = 'db_field_name',FIELD31,(TYPE=GROUP)
```
- ▶ The index specified with the -b flag *must not exist* in the data. So for example, if there is already an account index in the data that ACIF will extract, you should not specify -b "account", because although the code will currently permit you to do so, the results are unpredictable.
- ▶ You can specify IDX more than once in the -B flag (and input file name). However, the first IDX that appears in the file name is used, even though technically it is the second IDX. The code currently parses the file name in reverse because of the file extension.

So, for example:

- Input file name is:

MVS.12312006.test-CRD.12312007.00001.ARD

- -B option

"ign.IDX.ag.IDX.ign"

The value of the index loaded would be 12312006.

17.6 LDAP security

LDAP security has been embedded into the server code for non-z/OS integration. This will support authentication only. This requires the login to be placed in the user ID table.

17.6.1 Enable and disable LDAP authentication

To enable and disable LDAP authentication:

1. Starting from the Content Manager OnDemand Administration Client, right-click the Content Manager OnDemand server and select **System Parameters**, as shown in Figure 17-29.

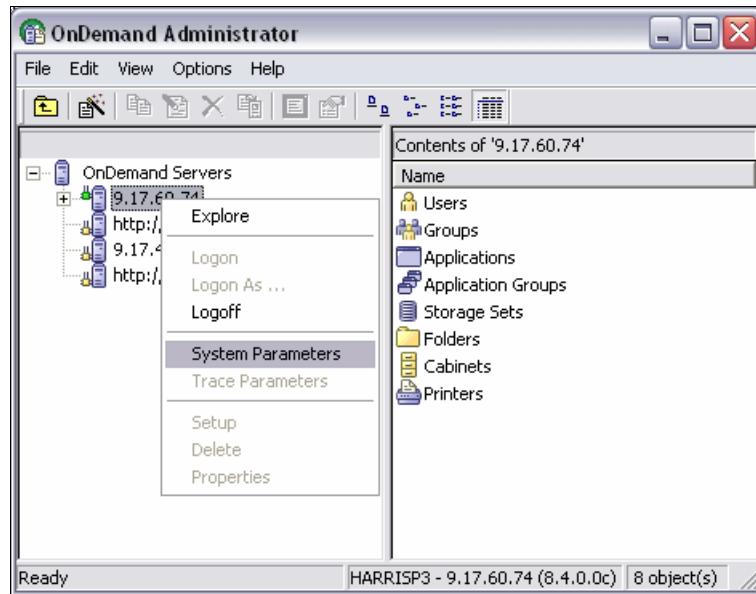


Figure 17-29 Accessing system parameters

2. In the System Parameters window, under LDAP Authentication, either select the **Enable LDAP** check box to enable LDAP or clear the check box to disable LDAP. See Figure 17-30. Save your settings.

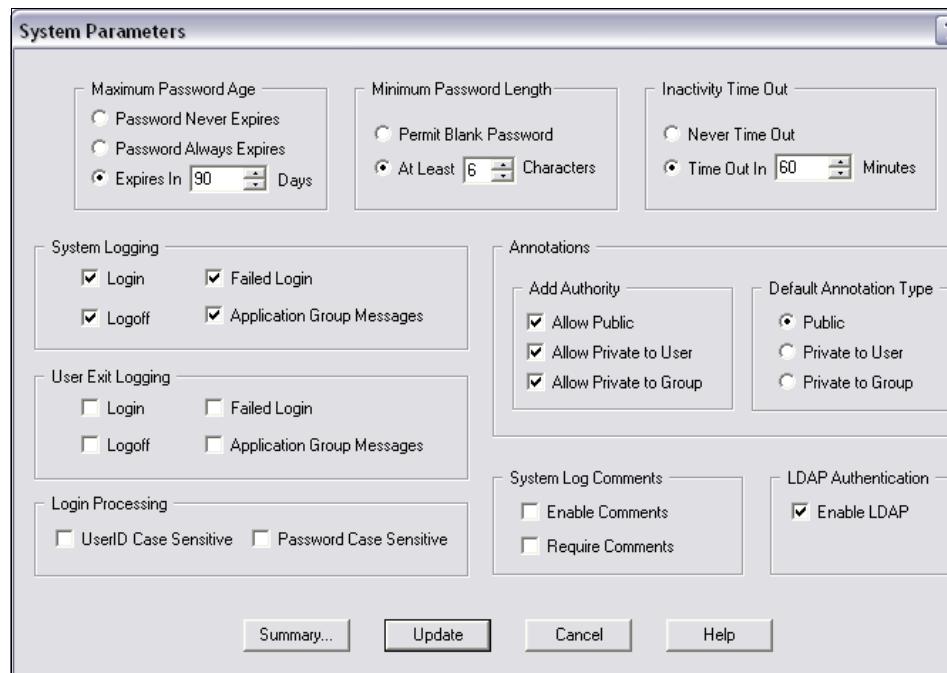


Figure 17-30 Enabling LDAP authentication

3. Define the name and port of the LDAP server you are going to authenticate to in the ARS.CFG with the parameters ARS_LDAP_SERVER and ARS_LDAP_PORT. See Figure 17-31.

```
#####
# LDAP Parameters (Library Server Only) #
#####
ARS_LDAP_SERVER=bluepages.ibm.com
ARS_LDAP_PORT=
ARS_LDAP_BASE_DN=ou=bluepages,o=ibm.com
ARS_LDAP_BIND_DN=
ARS_LDAP_BIND_DN_PWD=
ARS_LDAP_BIND_ATTRIBUTE=mail
ARS_LDAP_MAPPED_ATTRIBUTE=primaryuserid
ARS_LDAP_ALLOW_ANONYMOUS=TRUE
```

Figure 17-31 Sample ARS.CFG to configure LDAP authentication

17.7 64-bit support

Content Manager OnDemand V8.4 now has 64-bit support on all AIX, HP-UX, Solaris, Linux, and Linux for zSeries operating systems.

For a 64-bit Windows system, use the Windows Server 2003 R2 64-bit version. ODWEK will support both 32- and 64-bit.

The following are *only* supported on 32-bit systems:

- ▶ PDF Indexer
- ▶ Windows Server 2003 R2
- ▶ Windows Client
- ▶ Windows Administration Client
- ▶ z/OS

17.8 Tracing

Content Manager OnDemand has incorporated a trace facility into the code to help the support team perform problem determination. The trace facility is available in Content Manager OnDemand for Multiplatforms.

17.8.1 Configure tracing

To configure tracing, start from Content Manager OnDemand Administration Client:

1. Right-click the server name and select **Trace Parameter**. See Figure 17-32.

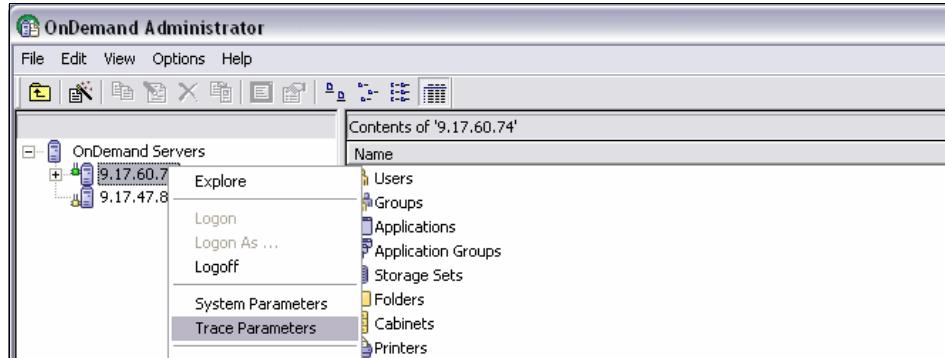


Figure 17-32 Selecting trace parameters

- At the System Trace Settings dialog box, you can specify the components you wish to trace. See Figure 17-33. By default, you can trace the database and the server by selecting the check box next to it. You can also specify the level of trace reporting and the level of logging for each component by selecting the check boxes under the Trace Level Reporting pane. Save your changes.

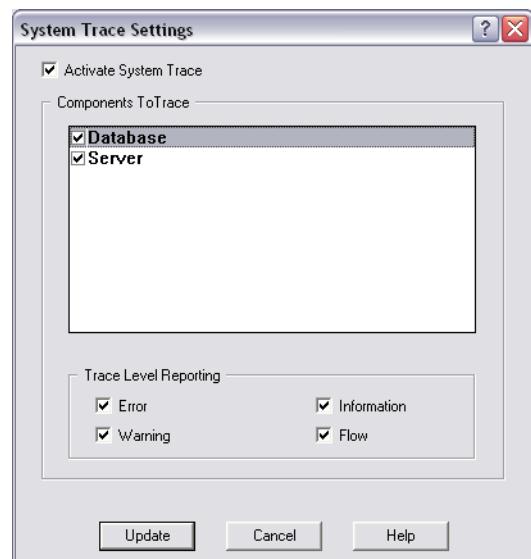


Figure 17-33 Selecting system trace settings

- The default trace log name is ARCHIVE.trace.log. The following line should be added to the ars.cfg for the directory path:

```
ARS_TRACE_SETTINGS=/usr/lpp/ars/config/trace.settings
```

Figure 17-34 shows a portion of the trace.settings file.

```
# By default the SRVR component will have ERROR, WARNING, and INFO
#
#####
#
# Parameter overrides for trace output
#
#####
#
# TRACE_FILE      will be created in the ARS_TMP directory
# TRACE_FORMAT    TEXT and CSV are the only outputs currently supported
# APPEND          0 create new file or 1 append to existing trace file
# MAX_LOG_SIZE    Max size of trace file in megabytes (0 = unlimited)
#
#####
#
[TRACE]
COMPONENT_LEVEL="000000000000000000070000000007"
TRACE_FILE=ARCHIVE.trace.log
TRACE_FORMAT=TEXT
APPEND=0
MAX_LOG_SIZE=0
```

Figure 17-34 Trace settings file

Figure 17-35 shows a sample trace.log file.

ID	PPID	TID	DATE	TIME	LEVEL	OUTPUT
429546	1491072	001B2099	06/25/07	14:36:45.251549	INFO	ArcTRACE_Init Tr
initiated						
454082	1	001F7067	06/25/07	14:36:45.275654	INFO	ArcCDB_Startup In
ndemand						
454082	1	001F7067	06/25/07	14:36:45.291891	INFO	ArcDBP_LoadDBMS
odule						
454082	1	001F7067	06/25/07	14:36:45.317799	INFO	ArcCDB_Init Conne
type=0,						
name=ARCHIVE)						
454082	1	001F7067	06/25/07	14:36:45.385492	INFO	ArcCDB_Init Conne
type=0,						
name=ARCHIVE)						
454082	1	001F7067	06/25/07	14:36:45.399411	INFO	ArcDBP_AppGroup
application						
group						
(ag_ws='System Log',						
agid=0,						
fid=0,						
ids='',						
get_field						
_intext=1)						
454082	1	001F7067	06/25/07	14:36:45.415420	INFO	ArcDBP_SplitIds
ids='',						
id=0,						
zero=0,						
optimize						
Only=0						
454082	1	001F7067	06/25/07	14:36:45.427401	INFO	ArcDBP_ReadOnly
454082	1	001F7067	06/25/07	14:36:45.439385	INFO	ArcDBP_SplitIds
bunts =						
0						
454082	1	001F7067	06/25/07	14:36:45.457896	INFO	ArcDBP_AppGroup
querying						
application						
group						
fields						
info						
(agid=5001,						
name='System Log')						
454082	1	001F7067	06/25/07	14:36:45.471368	INFO	ArcDBP_ReadOnly
optimize						
Only=0						
454082	1	001F7067	06/25/07	14:36:45.483942	INFO	ArcDBP_AppGroup
querying						
application						
group						
field						
alias						
(agid=5001,						
name='System Log',						
get_f						
optimize						
only=0						
454082	1	001F7067	06/25/07	14:36:45.511719	INFO	ArcDB_AppQuery

Figure 17-35 Sample log file



A

Additional material

This IBM Redbooks publication refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this IBM Redbooks publication is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG246915>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks publication form number, SG246915.

Using the Web material

The additional Web material that accompanies this IBM Redbooks publication includes the following files:

<i>File name</i>	<i>Description</i>
SG246915_StoreOD.zip	Store OnDemand
SG246915_ODToolbox.zip	OnDemand Toolbox

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	200 MB minimum
Operating System:	Windows
Processor:	Pentium® IV or higher
Memory:	512 MB

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Glossary

A

access. To obtain data from or to put data in storage.

ACIF. Advanced Function Presentation Conversion and Indexing Facility

Acrobat. The Adobe viewer for PDF files. Acrobat is similar to the IBM AFP Workbench, that is, a stand-alone viewer. Acrobat also supports a robust set of APIs. Through these APIs, Acrobat is integrated with the OnDemand client program.

active log file. The subset of files that consists of primary log files and secondary log files that are currently needed by the database manager for rollbacks and recovery.

active policy set. In Tivoli Storage Manager, the policy set within the policy domain that contains the most recently activated policy currently in use by all client nodes that have been assigned to that policy domain. See policy set.

active storage node. In a storage set, the storage node that is currently being used to load data.

adapter. A part that electrically or physically connects a device to a computer or to another device.

addressable point. Any point in a presentation surface that can be identified by a coordinate from the coordinate system of the presentation medium. See also picture element.

administrative client. (1) In OnDemand, the program that provides administrators with functions to maintain OnDemand groups, users, printers, applications, application groups, storage sets, and folders. (2) In Tivoli Storage Manager, the program that allows administrators to control and monitor the server through administrator commands.

ADSM. ADSTAR Distributed Storage Manager

ADSTAR Distributed Storage Manager. A program that provides storage management for archived files. See Tivoli Storage Manager.

Advanced Function Presentation (AFP). A set of licensed programs that use the all-points-addressable concept to print data on a wide variety of printers or display data on a variety of display devices. AFP also includes creating, formatting, archiving, viewing, retrieving, and distributing information.

Advanced Function Presentation application programming interface (AFP API). An AFP program shipped with PSF/MVS 2.1.1 and PSF/VM 2.1.1 that creates the AFP data stream from the COBOL and PL/1 high-level programming languages.

Advanced Function Presentation Conversion and Indexing Facility. A program shipped with OnDemand that you can use to convert a print file into a MO:DCA-P document, to retrieve resources used by the document, and to index the file for later retrieval and viewing.

Advanced Function Presentation data stream (AFP data stream). A presentation data stream that is processed in the AFP environment. MO:DCA-P is the strategic AFP interchange data stream. IPDS is the strategic AFP printer data stream.

AFP. Advanced Function Presentation

AFP API. Advanced Function Presentation application programming interface

AFPDS. A term formerly used to identify the composed page, MO:DCA-P-based data stream interchanged in AFP environments.

AIX. (1) Advanced Interactive Executive (2) The IBM version of the UNIX operating system.

AIX Acrobat Libraries. A subset of the Acrobat Libraries ported to AIX for use by OnDemand.

all-points-addressable (APA). The capability to address, reference, and position data elements at any addressable position in a presentation space or on a physical medium. An example of all points addressability is the positioning of text, graphics, and images at any addressable point on the physical medium. See also picture element.

all-points-addressable mode. A synonym for Page Mode.

alphabetic character. A letter or other symbol, excluding digits, used in a language. Usually the uppercase and lowercase letters A through Z plus other special symbols (such as \$ and _) allowed by a particular language. See also alphanumeric character.

alphanumeric character. Consists of letters, numbers, and often other symbols, such as punctuation marks and mathematical symbols. See also alphabetic character.

alphanumeric string. A sequence of characters consisting solely of the letters a through z and the numerals 0 through 9.

American National Standards Institute (ANSI). An organization for the purpose of establishing voluntary industry standards.

anchor point. The point in a document that signals to ACIF the beginning of a group of pages, after which it adds indexing structured fields to delineate this group.

ANSI. American National Standards Institute

ANSI carriage control character. A character that specifies that a write, space, or skip operation should be performed before printing the line containing the carriage control. ANSI carriage control characters are encoded in ASCII or EBCDIC.

APA. All points addressable

API. Application programming interface

application. In OnDemand, an object that describes the physical attributes of a report or input file, such as the type of data found in the input file, the code page, and whether the input data contains carriage control characters. An application also contains instructions that the data indexing and loading programs use to process the input data. Most customers define an application for each different output print data stream or source of data that they plan to store in OnDemand.

application group. A collection of one or more OnDemand applications that have similar indexing and storage management requirements. For example, two reports that can be retrieved using the same index fields and that are to be maintained by the system in the same storage locations for the same length of time can be placed in the same application group.

application programming interface (API). A formally defined programming language interface that is between a program and the user of a program.

archive copy group. In Tivoli Storage Manager, a policy object containing attributes that control the generation, destination, and expiration of archive files. An archive copy group belongs to a management class.

archive log file. The subject of files consisting of primary log files and secondary log files that are no longer needed for normal database processing.

archive media. Devices and volumes on which the long-term or backup copy of a report is stored. For example, an optical storage library is one type of archive media supported by OnDemand.

archive storage. The storage in which the long-term or backup copy of a report is maintained. Includes the devices and volumes on which the files are stored and the management policies that determine how long data is maintained in archive storage.

archive storage manager. The software product that manages archive media and maintains files in archive storage. See Tivoli Storage Manager.

ASCII (American Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

attachment. A device or feature attached to a processing unit, including required adapters. Contrast with Adapter.

authentication. The process of checking a user's password before allowing the user access to resources or the server.

authorize. (1) To grant to a user the right to communicate with or make use of a computer system or display station. (2) To give a user either complete or restricted access to an object, resource, or function.

B

BCOCA. Bar Code Object Content Architecture

backend. In the AIX operating system, the program that sends output to a particular device. Synonymous with backend program.

backend program. Synonym for backend.

Bar Code Object Content Architecture. An architected collection of control structures used to interchange and present barcode data.

bitmap. A file that contains a bit-mapped graphic.

BMP. Bitmap

byte. The amount of storage required to represent 1 character; a byte is 8 bits.

C

cache storage. The storage in which the primary or short-term copy of a report is stored. Usually disk storage. Most customers configure the system to maintain the most recent and frequently used versions of reports in cache storage.

carriage control character. The first character of an output record (line) that is to be printed; it determines how many lines should be skipped before the next line is printed.

case-sensitive. The ability to distinguish between uppercase and lowercase letters.

CCITT. Consultative Committee on International Telegraphy and Telephone

CD-ROM. Compact disc read-only memory

channel. A device connecting the processor to input and output devices.

channel adapter. A communication controller hardware unit used to attach the controller to a System/370™ data channel.

channel-attached. (1) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. (2) Synonymous with local.

character. A letter, digit, or other symbol that represents, organizes, or controls data.

character rotation. The alignment of a character with respect to its character baseline, measured in degrees in a clockwise direction. Examples are 0°, 90°, 180°, and 270°. Zero-degree character rotation exists when a character is in its customary alignment with the baseline.

character set. A group of characters used for a specific reason; for example, the set of characters a printer can print or a keyboard can support.

click. To press the left mouse button while pointing to an object such as a command button or a toolbar button.

client. (1) In a distributed file system environment, a system that is dependent on a server to provide it with programs or access to programs. (2) A personal computer connected to a network running OnDemand software that can log on and query the library server, retrieve documents from OnDemand, and view and print documents.

client domain. The set of optical drives and storage volumes used by Tivoli Storage Manager to store report files and resources belonging to an application group.

client node. An application group that has been registered to the Tivoli Storage Manager server.

COBOL. Common business-oriented language. A high-level programming language, based on English, that is used primarily for business applications.

code page. An ordered set of up to 256 predefined display symbols. The first 32 code points of each code page are reserved for control codes and are the same for all code pages, leaving up to 224 distinct display symbols per page.

Code Page Global Identifier (CPGID). A unique code page identifier that can be expressed as either a two-byte binary or a five-digit decimal value.

code point. A character within a code page.

coded font. An AFP font that associates a code page and a font character set.

command. A request to perform an operation or run a program. When parameters values, flags, or other operands are associated with a command, the resulting character string is a single command.

command line. The area of the screen where commands are displayed as they are typed.

communication method. The method used by OnDemand and Tivoli Storage Manager to exchange information.

communication protocol. A set of defined interfaces that allow computers to communicate with each other.

compact disc read-only memory (CD-ROM). High capacity read-only memory in the form of an optically read compact disk.

composed page. In Advanced Function Presentation, a page that can be printed only on an all-points-addressable output medium. It might contain composed text and raster images.

composed-text data file. A file containing text data and text control information that dictates the format, placement, and appearance of the data to be printed.

compression. A technique for removing strings of duplicate characters, gaps, empty fields, and trailing blanks before transmitting data.

concatenate. (1) To link together. (2) To join two character strings.

concatenated field. Two or more fields from a physical file record format that have been combined to make one field in a logical file record format.

conditional processing. A page definition function that allows input data records to partially control their own formatting.

configuration. The process of describing to a system the devices, optional features, and program products that are installed so that these features can be used. Contrast with customization.

configuration file. A file that specifies the characteristics of a system or subsystem; for example, the operating system queuing system.

configure. To describe to a system the devices, optional features, and licensed programs installed on a system.

console. The main operating system display station.

constant. A data item with a value that does not change during the running of a program. Contrast with variable.

Consultative Committee on International Telegraphy and Telephone (CCITT). A United Nations Specialized Standards group whose membership includes common carriers concerned with devising and proposing recommendations for international telecommunications representing alphabets, graphics, control information, and other fundamental information interchange issues.

Content Manager. A comprehensive set of Web-enabled, integrated software solutions from IBM for managing information and making it available to anyone, anywhere.

control character. A character that is not a graphic character such as a letter, number, or punctuation mark. Such characters are called control characters because they frequently act to control a peripheral device.

controller. A device that coordinates and controls the operation of one or more input/output devices, such as workstations, and synchronizes the operation of the system as a whole.

conversion. In programming languages, the transformation between values that represent the same data item but belong to different data types.

copies. See copy group.

copy group. In Tivoli Storage Manager, a policy object that contains attributes that control the generation, destination, and expiration of backup and archive files. There are two kinds of copy groups: backup and archive. Copy groups belong to management classes.

copy storage pool. A named collection of storage volumes that contains copies of files that reside in primary storage pools. Copy storage pools are used to back up the data stored in primary storage pools.

CPGID. Code Page Global Identifier

customization. The process of describing optional changes to defaults of a software program that is already installed on the system and configured so that it can be used. Contrast with configuration.

customize. To describe the system, the devices, programs, users, and user defaults for a particular data processing system or network. Contrast with configure.

D

daemon. In UNIX, a process begun by the root user or by the root shell that can be stopped only by the root user. Daemon processes generally provide services that must be available at all times, such as sending data to the printer. A daemon runs continuously, looking for work to do, performing that work, and waiting for more work. A daemon does not have a controlling terminal associated with it. The OnDemand data download program (ARSJESD) is an example of a daemon.

database. (1) The collection of information about all objects managed by OnDemand, including reports, groups, users, printers, application groups, storage sets, applications, and folders. (2) The collection of information about all objects managed by Tivoli Storage Manager, including policy management objects, administrators, and client nodes.

Database Managed Space (DMS). A type of DB2 tablespace. A DMS tablespace is managed by the database manager.

data set. Synonym for file.

data stream. A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form using a defined format.

data transfer. The movement, or copying, of data from one location and the storage of the data at another location.

data type. The type, format, or classification of a data object.

DCF. Document Composition Facility

decimal. Pertaining to a system of numbers to the base 10. The decimal digits range from 0 through 9.

decompression. A function that expands data to the length that preceded data compression. See also compression.

default. A value, attribute, or option that is assumed when no alternative is specified by the user.

default directory. The directory name supplied by the operating system if none is specified.

default printer. A printer that accepts all the printed output from a display station assigned to it.

default value. A value stored in the system that is used when no other value is specified. See also default.

desktop printer. In this publication, an IBM LaserPrinter 4019 or 4029, or compatible printer.

device class. A named group of Tivoli Storage Manager storage devices. Each device class has a unique name and represents a device type of disk, tape, or optical disk.

device driver. A program that operates a specific device, such as a printer, disk drive, or display.

device type. A type of Tivoli Storage Manager storage device. Each device class must be categorized with either the disk, tape, or optical disk devices types.

device-independent. Pertaining to a function that can be accomplished without regard for the characteristics of particular types of devices.

dialog box. An application window on the display that requests information from the user.

directory. (1) A type of file containing the names and controlling information for other files or directories. (2) A listing of related files arranged in a useful hierarchy.

disk operating system (DOS). An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

Distiller. A batch utility that converts PostScript files to Adobe PDF files. The distiller runs under AIX, HP-UX, Sun Solaris, and Windows servers.

DMS. Database Managed Space

document. (1) In OnDemand, a logical section of a larger file, such as an individual invoice within a report of thousands of invoices. A document can also represent an indexed group of pages from a report. (2) A file containing an AFP data stream document. An AFP data stream document is bounded by Begin Document and End Document structured fields and can be created using a text formatter such as Document Composition Facility (DCF).

Document Composition Facility (DCF). An IBM licensed program used to prepare printed documents.

domain. See policy domain or client domain.

DOS. Disk operating system

double-click. To rapidly press the left mouse button twice while pointing to an object.

download. To transfer data from one computer for use on another one. Typically, users download from a larger computer to a diskette or fixed disk on a smaller computer or from a system unit to an adapter.

drag. To hold down the left mouse button while moving the mouse.

driver. The end of a stream closest to an external interface. The principal functions of the driver are handling any associated device, and transforming data and information between the external device and stream.

E

EBCDIC. Extended Binary-Coded Decimal Interchange Code. This is the default type of data encoding in an MVS environment. Contrast with ASCII.

EIP. Enterprise Information Portal

enqueue. To place items in a queue.

enter. (1) An instruction to type specific information using the keyboard. (2) A keyboard key that, when pressed, confirms or initiates the selected command.

Enterprise Information Portal (EIP). An IBM software product that provides a coordinated, Web-enabled entry point to what is otherwise disconnected, incompatible data scattered across an enterprise.

environment variable. A variable that is included in the current software environment and is therefore available to any called program that requests it.

error condition. The state that results from an attempt to run instructions in a computer program that are not valid or that operate on data that is not valid.

error log. A file in a product or system where error information is stored for later access.

error log entry. In AIX, a record in the system error log describing a hardware or software failure and containing failure data captured at the time of the failure.

error message. An indication that an error has been detected.

error recovery. The process of correcting or bypassing the effects of a fault to restore a computer system to a prescribed condition.

error type. Identifies whether an error log entry is for a permanent failure, temporary failure, performance degradation, impending loss of availability, or undetermined failure.

ESS. IBM TotalStorage Enterprise Storage Server®

Ethernet. A 10-megabit baseband local area network using CSMA/CD (carrier sense multiple access with collision detection). The network allows multiple stations to access the medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission.

exit program. A user-written program that is given control during operation of a system function.

exit routine. A routine that receives control when a specified event occurs, such as an error.

expiration. The process of deleting index data and reports based on storage management information. The OnDemand database manager and the storage managers run expiration processing to remove data that is no longer needed from storage volumes and reclaim the space.

Extended Binary-Coded Decimal Interchange Code (EBCDIC). A coded character set consisting of eight-bit coded characters.

external library resource (member). Objects that can be used by other program products while running print jobs; for example, coded fonts, code pages, font character sets, form definitions, page definitions, and page segments. Synonym for resource object.

F

FCB. Forms control buffer

field. A specified area in a record used for a particular type of data; for example, a group of characters that represent a customer's name.

file. (1) A named set of records stored or processed as a unit. (2) The major unit of data storage and retrieval. A file consists of a collection of data in one of several prescribed arrangements and described by control information to which the operating system has access.

file system. The collection of files and file management structures on a physical or logical mass storage device, such as a diskette or a minidisk.

file transfer. In remote communications, the transfer of a file or files from one system to another over a communications link.

File Transfer Protocol (FTP). In TCP/IP, the protocol that makes it possible to transfer data among hosts and to use foreign hosts indirectly.

fixed disk. A flat, circular, nonremovable plate with a magnetizable surface layer on which data can be stored by magnetic recording. A rigid magnetic disk.

fixed-disk drive. The mechanism used to read and write information on a fixed disk.

folder. In OnDemand, the end-user view of data stored in the system. Folders provide users a convenient way to find related information, regardless of the source of the information or where the data is stored.

font. (1) A family of characters of a given size and style, for example 9-point Helvetica. (2) A set of characters in a particular style. See raster font.

font character set. Part of an AFP font that contains the raster patterns, identifiers, and descriptions of characters. Often synonymous with Character Set. See also coded font.

form definition (FORMDEF). A form definition is a resource used by OnDemand. A form definition specifies the number of copies to be printed, whether the sheet should be printed on both sides, the position of a page of data on the sheet, text suppression, and overlays to be used (if any). Synonymous with FORMDEF.

FORMDEF. Form Definition

FSA. Functional SubSystem Application. A collection of programs residing in the FSS address space that control a device.

FSI. Functional SubSystem Interface. An MVS or OS/390 interface that allows communication between JES and a FSS and FSS applications. Download uses an FSI to communicate with the operating system and JES to process spool data sets created by application programs.

FSS. Functional SubSystem. An MVS or OS/390 subsystem comprised of programs residing in the same address space that provide JES-related functions. For example, print programs that extend the scope of JES processing can be defined as an FSS.

FTP. File Transfer Protocol

G

GB. Gigabyte

GIF. Graphic Interchange Format

gigabyte. A unit of memory or space measurement equal to approximately one billion bytes. One gigabyte equals 1,000 megabytes.

GOCA. Graphic Object Content Architecture

graphic. A symbol produced by a process such as handwriting, drawing, or printing.

graphic character. A character that can be displayed or printed.

Graphic Object Content Architecture. An architecture that provides a collection of graphics values and control structures used to interchange and present graphics data.

graphical user interface (GUI). A type of user interface that takes advantage of a high-resolution monitor, including some combination of graphics, the use of pointing devices, menu bars, overlapping windows, and icons.

graphics. A type of data created from such fundamental drawing units such as lines, curves, polygons, and so forth.

Graphic Interchange Format (GIF). A bit-mapped color graphics file format for IBM and IBM-compatible computers. GIF employs an efficient compression technique for high resolution graphics.

group. (1) A named collection of sequential pages that form a logical subset of a document. (2) A named collection of users assigned a specific role on the system or belonging to a specific department.

GUI. Graphical user interface

H

hardware. The physical equipment of computing and computer-directed activities. The physical components of a computer system. Contrast with software.

help. One or more files of information that describe how to use application software or how to perform a system function.

hex. Hexadecimal

hexadecimal (hex). Pertaining to a system of numbers in the base sixteen; hexadecimal digits range from 0 (zero) through 9 (nine) and A (ten) through F (fifteen).

host. (1) The primary or controlling computer in the communications network. (2) See host system.

host-based computer. (1) In a computer network a computer that provides end users with services such as computation and databases and that usually performs network control functions. (2) The primary or controlling computer in a multiple-computer installation.

host system. (1) The controlling or highest level system in a data communication configuration, for example, an OS/390 system is the host system for the terminals connected to it. (2) In TCP/IP, a computer that is a peer system in a network.

I

IBM TotalStorage Enterprise Storage Server (ESS). An IBM disk storage system that provides industry-leading availability, performance, manageability, and scalability. Virtually all types of servers can concurrently attach to the Enterprise Storage Server, including S/390, UNIX servers, and Windows servers. As a result, the Enterprise Storage Server is ideal for organizations with growing e-business operations that are being handled by multiple heterogeneous servers.

icon. A 32 by 32 pixel bitmap used by the windows manager to represent an application or other window.

image. (1) An electronic representation of a picture produced by means of sensing light, sound, electron radiation, or other emanations coming from the picture or reflected by the picture. An image can also be generated directly by software without reference to an existing picture. (2) An electronic representation of an original document recorded by a scanning device.

Image Object Content Architecture. An architected collection of constructs used to interchange and present images.

index. (1) A process of segmenting a print file into uniquely identifiable groups of pages (a named collection of sequential pages) for later retrieval. (2) A process of matching reference points within a file and creating structured field tags within the MO:DCA-P document and the separate index object file.

index object file. An index-information file created by ACIF that contains the Index Element (IEL) structured fields, which identify the location of tagged groups in the AFP file. The indexing tags are contained in the Tagged Logical Element (TLE) structured fields.

indexing. (1) A process of segmenting a print file into uniquely identifiable groups of pages (a named collection of sequential pages) for later retrieval. (2) In ACIF, a process of matching reference points within a file and creating structured field tags within the MO:DCA-P document and the separate index object file.

indexing with data values. Adding indexing tags to a MO:DCA-P document using data that is already in the document and that is consistently located in the same place in each group of pages.

indexing with literal values. Adding indexing tags to a MO:DCA-P document by assigning literal values as indexing tags, because the document is not organized such that common data is located consistently throughout the document.

InfoPrint Manager. A sophisticated IBM print subsystem that drives AFP printers, PostScript printers, and PCL printers. InfoPrint Manager is supported under AIX, OS/390, Windows NT, and Windows 2000. InfoPrint Manager manages printer resources such as fonts, images, electronic forms, form definitions, and page definitions, and provides error recovery for print jobs. When printing line data, InfoPrint Manager supports external formatting using page definitions and form definitions. This external formatting extends page printer functions such as electronic forms and use of typographic fonts without any change to applications that generate the data.

informational message. (1) A message that provides information to the end-user or system administrator but does not require a response. (2) A message that is not the result of an error condition.

input file. A file opened in order to allow records to be read.

install. (1) To add a program, program option, or software program to the system in a manner such that it might be executed and will interact properly with all affected programs in the system. (2) To connect a piece of hardware to the processor.

intelligent printer data stream (IPDS). An all-points-addressable data stream that allows users to position text, images, and graphics at any defined point on a printed page.

interface. Hardware, software, or both, that links systems, programs, or devices.

Internet. A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the protocol for transmitting information.

Internet Protocol (IP). In TCP/IP, a protocol that routes data from its source to its destination in an Internet environment.

IOCA. Image Object Content Architecture

IP. Internet Protocol

IPDS. Intelligent printer data stream

J

job. One or more related procedures or programs grouped into a procedure, identified by appropriate job control statements.

job queue. A list of jobs waiting to be processed by the system.

Joint Photographic Experts Group (JPEG). An image compression standard developed to handle larger images with many colors. JPEG uses a “lossy” algorithm, which means there is some loss of detail when saving and viewing images in this format. However, JPEG files can offer as much as 35% improvement in file size and compression.

JPEG. Joint Photographic Experts Group

K

kernel. The part of an operating system that performs basic functions such as allocating hardware resources.

kernel extension. A program that modifies parts of the kernel that can be customized to provide additional services and calls. See kernel.

K-byte. Kilobyte

keyword. Part of a command operand that consists of a specific character string.

kilobyte (K-byte). 1024 bytes in decimal notation when referring to memory capacity; in all other cases, it is defined as 1000.

L

LAN. Local area network

LAN server. A data station that provides services to other data stations on a local area network; for example, file server, print server, mail server.

laser printer. A nonimpact printer that creates, by means of a laser beam directed on a photosensitive surface, a latent image which is then made visible by toner and transferred and fixed on paper.

Lempel Ziv Welsh (LZW). A data compression algorithm. OnDemand uses the 16-bit version of LZW to compress data.

library. System storage for generated form definitions and page definitions.

library resource (member). A named collection of records or statements in a library.

library resource name. A name by which an object might be called from a library by AFP as part of a print job. Includes the 2-character prefix for the type of object, such as P1 for page definitions, F1 for form definitions, or O1 for overlays (also known as resource name).

library server. In OnDemand, the workstation or node that users must go through to access the system. The library server controls the OnDemand database.

licensed program. A separately priced program and its associated materials that bear a copyright and are offered to customers under the terms and conditions of a licensing agreement.

line data. Data prepared for printing on a line printer, such as an IBM 3800 Model 1 Printing Subsystem. Line data is usually characterized by carriage-control characters and table reference characters.

line-data print file. A file that consists of line data, optionally supplemented by a limited set of structured fields.

line printer. A device that prints a line of characters as a unit. Contrast with page printer.

line printer daemon (LPD). In TCP/IP, the command responsible for sending data from the spooling directory to a printer.

line printer requestor (LPR). In TCP/IP, a client command that allows the local host to submit a file to be printed on a remote print server.

literal. (1) A symbol or a quantity in a source program that is itself data, rather than a reference to data. (2) A character string whose value is given by the characters themselves; for example, the numeric literal 7 has the value 7, and the character literal CHARACTERS has the value CHARACTERS.

loading. The logical process of archiving reports in OnDemand. During the loading process, OnDemand processes reports, creates index data, and copies report data and resources to cache storage and archive storage.

local. Pertaining to a device accessed directly without use of a telecommunication line.

local area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary might be subject to some form of regulation. (2) A network in which a set of devices is connected to one another for communication and that can be connected to a larger network. See also token-ring network.

logical page. In the IMS™ message format service, a user-defined group of related message segment and field definitions.

logical volume. The combined space from all volumes defined to either the Tivoli Storage Manager database or recovery log. The database resides on one logical volume and the recovery log resides on a different logical volume.

log file. A fixed-length file used to record changes to a database.

LPD Line printer daemon

LPR. Line printer requestor.

LZW. See Lempel Ziv Welsh

M

M byte. Megabyte

MB. Megabyte

machine carriage control character. A character that specifies that a write, space, or skip operation should be performed either immediately or after printing the line containing the carriage control.

mainframe. A large computer, particularly one to which other computers can be connected so that they can share facilities the mainframe provides. The term usually refers to hardware only.

management class. A logical area of storage that is managed by Tivoli Storage Manager. A management class is a policy object that is a named collection of copy groups. A management class can contain one backup copy group, one archive copy group, a backup and archive copy group, or zero copy groups. Users can bind each file to a management class to specify how the server should manage backup versions or archive copies of files. See copy group.

mapping. (1) A list that establishes a correspondence between items in two groups. (2) The process of linking database fields in an application group to folder search and display fields.

megabyte (MB). When used with hard drive, diskette, or removable media storage capacity, 1000000 bytes. When referring to system memory capacity, 1048576 bytes.

memory. Program-addressable memory from which instructions and other data can be loaded directly into registers for subsequent running or processing. Memory is sometimes referred to as storage.

menu bar. The area at the top of a window that contains choices that give a user access to actions available in that window.

message. Information from the system that informs the user of a condition that might affect further processing of a current program.

migration. (1) The process of moving data from one computer system to another without converting the data. (2) The process of moving report files, resources, and index data from cache storage to long-term (optical or tape) storage.

mirroring. In Tivoli Storage Manager, a feature that protects against data loss with the database or recovery log by writing the same data to multiple disks at the same time. Mirroring supports up to three exact copies of each database or recovery log.

Mixed Object Document Content Architecture -Presentation (MO:DCA-P). (1) A strategic, architected, device-independent data stream for interchanging documents. (2) A printing data stream that is a subset of the Advanced Function Presentation data stream.

MO:DCA-P. Mixed Object: Document Content Architecture for Presentation

mount. To make a file system accessible.

mouse. A hand-held locator that a user operates by moving it on a flat surface. It allows the user to select objects and scroll the display panel by pressing buttons.

N

network. A collection of data processing products that are connected by communication lines for information exchange between locations.

Network File System (NFS). A protocol developed by Sun Microsystems that uses Internet Protocol to allow a set of cooperating computers to access each other's file system as though they were local.

NFS. Network File System

node. A workstation that operates as an OnDemand library server or object server and is connected to a TCP/IP network.

notes. Electronic comments, clarifications, and reminders that can be attached to an OnDemand document.

non-IPDS printer. In this publication, a printer that is not channel-attached and which does not accept the Intelligent Printer Data Stream.

numeric. Pertaining to any of the digits 0 through 9.

O

object. (1) A collection of structured fields. The first structured field provides a begin-object function and the last structured field provides an end-object function. The object might contain one or more other structured fields whose content consists of one or more data elements of a particular data type. An object might be assigned a name, which might be used to reference the object. Examples of objects are text, graphics, and image objects. (2) A resource or a sequence of structured fields contained within a larger entity, such as a page segment or a composed page. (3) A collection of data referred to by a single name.

object server. In OnDemand, a workstation or node controlled by a storage manager to maintain reports in cache storage, and optionally, archive storage.

offset. The number of measuring units from an arbitrary starting point in a record, area, or control block to some other point.

online. Being controlled directly by or directly communicating with the computer.

operating environment. (1) The physical environment; for example, temperature, humidity, and layout. (2) All of the basic functions and the user programs that can be executed by a store controller to enable the devices in the system to perform specific operations. (3) The collection of store controller data, user programs, lists, tables, control blocks, and files that reside in a subsystem store controller and control its operation.

operating system. Software that controls the running of programs and that also can provide such services as resource allocation, scheduling, input and output control, and data management.

optical library. A storage device that houses optical disk drives and optical disks, and contains a mechanism for moving optical disks between a storage area and optical disk drives.

optimize. To improve the speed of a program or to reduce the use of storage during processing.

outline font. (1) Font whose graphic character shapes are defined as mathematical equations rather than by raster patterns. (2) Font created in the format described in *Adobe Type 1 Font Format*, a publication available from Adobe Systems, Inc. Synonymous with Type 1 fonts.

overlay. A collection of predefined, constant data such as lines, shading, text, boxes, or logos, that is electronically composed and stored as an AFP resource file that can be merged with variable data on a page while printing or viewing.

P

page. (1) A collection of data that can be printed on one side of a sheet of paper or a form. (2) The boundary for determining the limits of printing. See also logical page and physical page. (3) Part of an AFP document bracketed by a pair of Begin Page and End Page structured fields.

page definition. A resource used by OnDemand that defines the rules of transforming line data into composed pages and text controls.

page printer. A device that prints one page as a unit. Contrast with line printer.

page segment. In Advanced Function Presentation, a resource that can contain text and images and can be positioned on any addressable point on a page or an electronic overlay.

PAGEDEF. Page definition

parallel device. A device that can perform two or more concurrent activities. Contrast with serial device.

parameter. (1) Information that the user supplies to a panel, command, or function. (2) In the AIX operating system, a keyword-value pair.

partitioned data set. A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

path. In a network, any route between any two nodes.

path name. A name that specifies the location of a directory within a file system. Path names are used to locate and reference directories and their contents.

PC. Personal Computer

PCL. Printer Control Language

PCX. Picture Exchange Format

PDF. Portable Document Format

permissions. Codes that determine the users that can access a system, that determine how data can be used by any users who can access the system, and that determine other types of tasks users of the system can perform.

personal computer. A microcomputer primarily intended for stand-alone use by an individual.

physical page. In the IMS message format service, all or part of a logical page that is to be entered or displayed at one time.

picture element (pel). The smallest printable or displayable unit that can be displayed. A common measurement of device resolution is picture elements per inch.

Picture Exchange Format (PCX). A file that contains a graphic in the PCX graphics file format, which was originally developed for the PC Paintbrush program, but is now widely used by other programs.

pobe. The printer input/output back end program used by AIX for printing tasks.

pipe. To direct the data so that the output from one process becomes the input to another process. The standard output of one command can be connected to the standard input of another with the pipe operator (`|`). Two commands connected in this way constitute a pipeline.

point. (1) To move the mouse pointer to a specific object. (2) A unit of typesetting measure equal to 0.01384 inch (0.35054 mm), or about one-seventy second of an inch. There are 12 points per pica.

point size. The height of a font in points. See also point.

policy domain. In Tivoli Storage Manager, a policy object that contains policy sets, management classes, and copy groups that is used by a group of client nodes. See policy set, management class, copy group, and client node.

policy set. In Tivoli Storage Manager, a policy object that contains a group of management class definitions that exist for a policy domain. At any one time, there can be many policy sets within a policy domain but only one policy set can be active. See management class and active policy set.

port. (1) A part of the system unit or remote controller to which cables for external devices (display stations, terminals, or printers) are attached. The port is an access point for data entry or exit. (2) A specific communications endpoint within a host. A port is identified by a port number.

Portable Document Format (PDF). A distilled version of PostScript data that adds structure and efficiency. PDF data has the same imaging model as PostScript but does not have its programmability. PDF also provides direct access to pages and allows hypertext links, bookmarks, and other navigational aids required for viewing. The text in a PDF file is usually compressed using LZW methods. The images in a PDF file are usually compressed using CCITT or JPEG methods.

PostScript. Adobe's page description language used for printing. PostScript is a flexible programming language and imaging model but is not as structured as AFP. PostScript cannot be parsed to determine page boundaries, it must be interpreted. Because of this limitation, PostScript is not practical for archiving and viewing. Adobe created PDF for archiving and viewing.

press. To touch a specific key on the keyboard.

primary log file. A set of one or more log files used to record changes to a database. Storage for these files is allocated in advance.

primary storage pool. A named collection of storage volumes in which Tivoli Storage Manager stores archive copies of files.

print file. (1) The output of a user-defined program that is to be indexed and loaded into the system. (2) A file that a user wants to print.

print job. A series of print files scheduled for printing. At print submission time, the user can request one or more files to be printed; therefore, a print job consists of one or more print files.

print queue. A file containing a list of the names of files waiting to be printed.

Print Services Facility (PSF). A sophisticated IBM print subsystem that drives IPDS page printers. PSF is supported under MVS, VSE, VM, OS/2®, AIX, and is a standard part of the operating system under OS/400. PSF manages printer resources such as fonts, images, electronic forms, form definitions, and page definitions, and provides error recovery for print jobs.

When printing line data, PSF supports external formatting using page definitions and form definitions. This external formatting extends page printer functions such as electronic forms and use of typographic fonts without any change to applications that generate the data.

Print Services Facility/2 (PSF/2). PSF/2 is an OS/2-based print server that drives IPDS page printers, as well as IBM PPDS and HP-PCL compatible printers. PSF/2 manages printer resources and provides error recovery for print jobs. PSF/2 supports distributed printing of AFP print jobs from PSF for AIX, PSF/MVS, PSF/VSE, PSF/VM, and OS/400. PSF/2 also supports printing from a wide range of workstation applications, including Microsoft Windows and OS/2 Presentation Manager, as well as the ASCII, PostScript, and AFP data streams.

Print Services Facility for AIX (PSF for AIX). An IBM licensed program that produces printer commands from the data sent to it and it runs on the AIX/6000 operating system.

print spooler. The print spooler directs the printing of data from different applications. It temporarily stores information in separate files until they are printed.

Printer Control Language (PCL). The data stream used by Hewlett-Packard LaserJet II and III and other compatible printers.

process. An activity within the system that is started, such as a command, a shell program, or another process.

profile. (1) A file containing customized settings for a system or user. (2) Data describing the significant features of a user, program, or device.

program level. The version, release, modification, and fix levels of a program.

prompt. A displayed symbol or message that requests information or operator action.

protocol. A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.

PSF. Print Services Facility

PSF/2. Print Services Facility/2

PSF for AIX. Print Services Facility for AIX

PTF. Program temporary fix

Q

qdaemon. The daemon process that maintains a list of outstanding jobs and sends them to the specified device at the appropriate time.

qualified name. (1) A data name explicitly accompanied by a specification of the class to which it belongs in a specified classification system. (2) A name that has been made unique by the addition of one or more qualifiers.

queue. (1) A line or list formed by items waiting to be processed. (2) To form or arrange in a queue.

queue device. A logical device defining characteristics of a physical device attached to a queue.

R

radio button. Round option buttons grouped in windows; one is preselected. Like a radio in an automobile, select only one button (“station”) at a time.

RAM. Random access memory. Specifically, the memory used for system memory. Sometimes this memory is referred to as main storage.

raster. In Advanced Function Presentation, an on/off pattern of electrostatic images produced by the laser print head under control of the character generator.

raster font. A font in which the characters are defined directly by the raster bit map. See font. Contrast with outline font.

raster graphics. Computer graphics in which a display image is composed of an array of pixels arranged in rows and columns.

read access. In computer security, permission to read information.

record. (1) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. (2) A set of data treated as a unit. (3) A collection of fields treated as a unit.

recovery log. In Tivoli Storage Manager, a log of updates that are about to be written to the database. The log can be used to recover from system and media failures.

recovery procedure. (1) An action performed by the operator when an error message appears on the display panel. This action usually permits the program to run the next job. (2) The method of returning the system to the point where a major system error occurred and running the recent critical jobs again.

register. To define a client node to Tivoli Storage Manager.

remote. Pertaining to a system or device that is accessed through a communications line. Contrast with Local.

remote print. Issuing print jobs to one machine (client) to print on another machine (server) on a network.

remote system. A system that is connected to your system through a communication line.

report. A print data stream produced by a user-defined program or other software program that can contain hundreds or thousands of pages of related information. Most reports can be logically divided and indexed into single and multiple page objects called documents.

resolution. (1) In computer graphics, a measure of the sharpness of an image, expressed as the number of lines and columns on the display panel. (2) The number of pels per unit of linear measure.

resource. A collection of printing instructions, and sometimes data to be printed, that consists entirely of structured fields. A resource can be stored as a member of a directory and can be called for by the Print Services Facility when needed. The different resources are: coded font, character set, code page, page segment, overlay, and form definition.

resource directory. A place in which resource files are stored.

resource management. The function that protects serially accessed resources from concurrent access by computing tasks.

retention. The amount of time, in days, that archived files will be retained in Tivoli Storage Manager before they are deleted.

retry. To try the operation that caused the device error message again.

return code. (1) A value that is returned to a program to indicate the results of an operation issued by that program. (2) A code used to influence the running of succeeding instructions.

root. On UNIX servers, the user name for the system user with the most authority.

root file system. In UNIX environments, the file system that contains all of the default installation and program directories in the system.

root user. In UNIX environments, an expert user who can log in and execute restricted commands, shut down the system, and edit or delete protected files.

root volume group. In UNIX environments, the volume group, identified with a single / (forward slash) that contains all the directories in the root file system.

rotation. (1) The alignment of a character with respect to its character baseline, measured in degrees in a clockwise rotation. Examples are 0°, 90°, 180°, and 270°. Zero-degree character rotation exists when a character is in its customary alignment with the baseline. Synonymous with Character Rotation. (2) The number of degrees a character is turned relative to the page coordinates. (3) The orientation of the characters of a font with respect to the baseline.

routing. The assignment of the path by which a message will reach its destination.

S

secondary log file. A set of one or more log files used to record changes to a database. Storage for these files is allocated as needed when the primary log fills up.

segment. (1) A collection of composed text and images, prepared before formatting and included in a document when it is printed. See Page Segment. (2) The resource that contains the structured-field definition of a page segment. (3) A 100 page portion of a report file. OnDemand divides report files into segments to provide enhanced performance and maintenance.

segment table. A high-level index to index data stored in an application group. Each row in the segment table identifies a table of application group index data. OnDemand uses the segment table to limit a query to a specific table of application group index data.

select. To choose a menu command or other object with a single click of the mouse.

serial device. A device that performs functions sequentially, such as a serial printer that prints one byte at a time. Contrast with parallel device.

server. (1) On a network, the computer that contains the data or provides the facilities to be accessed by other computers in the network. (2) A program that handles protocol, queuing, routing, and other tasks necessary for data transfer between devices in a computer system. (3) A workstation connected to a TCP/IP network that runs the OnDemand programs that store, retrieve, and maintain report files. OnDemand supports two types of servers: a library server and an object server.

server options file. The Tivoli Storage Manager file that specifies processing options for communication methods, tape handling, pool sizes, language, and date, time, and number formats.

shell. In UNIX environments, a software interface between a user and the operating system of a computer. Shell programs interpret commands and user interactions on devices such as keyboards and pointing devices and communicate them to the operating system.

skip-to-channel control. A line printer control appearing in line data. Allows space to be left between print lines. Compatible with page printers when the data is formatted by page definitions.

SMIT. System Management Interface Tool

SMS. System Managed Space

software. Programs, procedures, rules, and any associated documentation pertaining to the operating of a system. Contrast with hardware.

spool file. (1) A disk file containing output that has been saved for later printing. (2) Files used in the transmission of data among devices. spooling (simultaneous peripheral operation).

spooling subsystem. A synonym for the queuing system that pertains to its use for queuing print jobs.

stand-alone workstation. A workstation that can perform tasks without being connected to other resources such as servers or host systems.

standard input. The primary source of data going into a command. Standard input comes from the keyboard unless redirection or piping is used, in which case standard input can be from a file or the output from another command.

standard output. The primary destination of data coming from a command. Standard output goes to the display unless redirection or piping is used, in which case standard output can be to a file or another command.

status. (1) The current condition or state of a program or device. For example, the status of a printer. (2) The condition of the hardware or software, usually represented in a status code.

storage. (1) The location of saved information. (2) In contrast to memory, the saving of information on physical devices such as disk or tape.

storage device. A functional unit for storing and retrieving data.

storage hierarchy. A logical ordering of storage devices. Generally, the ordering is based on the speed and capacity of the devices.

storage node. A named object that identifies the locations used to hold report data. A storage node can identify cache storage and a Tivoli Storage Manager domain on an OnDemand object server.

storage object. A portion of a storage volume managed as a single entity. A storage object can contain many segments of report data.

storage pool. In Tivoli Storage Manager, a named collection of storage volumes that is the destination for archived files.

storage pool volume. In Tivoli Storage Manager, a volume that has been assigned to a storage pool to store archived files.

storage set. A named collection of storage nodes that determines the locations that can hold report data.

storage volume. A volume that has been assigned to hold report data on an OnDemand server.

string. A series or set of alphabetic or numeric characters. A string can be composed of letters, numbers, and special characters.

structure. A variable that contains an ordered group of data objects. Unlike an array, the data objects within a structure can have varied data types.

structured field. (1) A self-identifying, variable-length, bounded record that can have a content portion that provides control information, data, or both. (2) A mechanism that permits variable length data to be encoded for transmission in the data stream. See field.

subdirectory. In the file system hierarchy, a directory contained within another directory.

subroutine. (1) A sequenced set of statements or coded instructions that can be used in one or more computer programs and at one or more points in a computer program. (2) A routine that can be part of another routine.

syntax. The grammatical rules for constructing a command, statement, or program.

syntax diagram. A diagram for a command that displays how to enter the command on the command line.

system console. A console, usually equipped with a keyboard and display panel, that is used by an operator to control and communicate with a system. Synonymous with console.

system customization. Specifying the devices, programs, and users for a particular data processing system. See also configuration.

system integrity. In computer security, the quality of a system that can perform its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

System Managed Space (SMS). A type of DB2 tablespace. An SMS tablespace is managed by the filesystem manager.

system management. The tasks involved in maintaining the system in good working order and modifying the system to meet changing requirements.

System Management Interface Tool (SMIT). In the AIX operating system, a series of panels that allow you to perform system functions without directly issuing any commands.

system memory. Synonymous with Main Storage, but used in hardware to refer to semiconductor memory (modules).

system prompt. Synonym for command line. The system prompt is the symbol that appears at the command line of an operating system. The system prompt indicates that the operating system is ready for the user to enter a command.

T

table. A named collection of data consisting of rows and columns.

table reference character (TRC). (1) Usually, the second byte on a line in the user's data. This byte contains a value (0–126) that is used to select a font to be used to print that line. (2) In the 3800 Printing Subsystem, a numeric character (0, 1, 2, or 3) corresponding to the order in which the character arrangement table names have been specified with the CHARS keyword. It is used for selection of a character arrangement table during printing.

tablespace. An abstraction of a collection of containers into which database objects are stored. A tablespace provides a level of indirection between a database and the tables stored within the database. A tablespace has space on media storage devices assigned to it and has tables created within it.

tag. (1) A type of structured field used for indexing in an AFP document. Tags associate an index attribute-value pair with a specific page or group of pages in a document. (2) In text formatting markup language, a name for a type of document element that is entered in the source document to identify it.

Tagged Image File Format (TIFF). A bit-mapped graphics format for scanned images with resolutions of up to 300 dpi. TIFF simulates gray scale shading.

TB. Terabyte

TCP. Transmission Control Protocol

TCP/IP. Transmission Control Protocol/Internet Protocol

terabyte. A unit of memory or space measurement capacity equal to approximately one trillion bytes. One terabyte is equal to 1,000 gigabytes, or one million megabytes.

text. (1) A type of data consisting of a set of linguistic characters (letters, numbers, and symbols) and formatting controls. (2) In word processing, information intended for human viewing that is presented in a two-dimensional form, such as data printed on paper or displayed on a panel.

throughput. A measure of the amount of work performed by a computer system over a period of time, for example, the number of jobs per day.

TIFF. Tagged Image File Format

Tivoli Storage Manager. An IBM software program that provides archive storage management of data stored in an OnDemand system.

token name. An eight-byte name that can be given to all data stream objects.

token-ring network. A ring network that allows unidirectional data transmission between data stations, by a token passing procedure, such that the transmitted data return to the transmitting station. (T)

toolbar. The region directly beneath the menu bar of the main window in OnDemand client programs that support a graphical user interface.

toolbar button. A small bitmap on the toolbar that represents a command in OnDemand client programs that support a graphical user interface. Click a toolbar button to quickly access a command.

transfer. To send data to one place and to receive data at another place.

transform. To change the form of data according to specified rules without significantly changing the meaning of the data.

Transmission Control Protocol (TCP). A communications protocol used in Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. TCP provides a host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

TRC. Table reference character

trigger. Data values that ACIF searches for in the input data stream, to delineate the beginning of a new group of pages. The first trigger is then the anchor point that ACIF uses to locate index values.

Tivoli Storage Manager. Tivoli Storage Manager

type. To enter specific information using the keyboard, typing characters exactly as given.

U

unformatted print data. Data that is not formatted for printing. A page definition can contain controls that map unformatted print data to its output format.

UNIX operating system. An operating system developed by Bell Laboratories that features multiprogramming in a multi-user environment. The UNIX operating system was originally developed for use on minicomputers but has been adapted for mainframes and microcomputers.

upload. To transfer data from one computer to another. Typically, users upload from a small computer to a large one.

user. A person authorized to logon to an OnDemand server.

user exit. (1) A point in an IBM-supplied program at which a user-defined program might be given control. (2) A programming service provided by an IBM software product that might be requested during the execution of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

user interface. The hardware, software, or both that implements a user interface, allowing the user to interact with and perform operations on a system, program, or device. Examples are a keyboard, mouse, command language, or windowing subsystem.

V

value. (1) A set of characters or a quantity associated with a parameter or name. (2) A quantity assigned to a constant, variable, parameter, or symbol.

variable. (1) A name used to represent a data item whose value can change while the program is running. (2) In programming languages, a language object that can take different values at different times. (3) A quantity that can assume any of a given set of values.

version number. The version level of a program, which is an indicator of the hardware and basic operating system upon which the program operates. The version, release, modification, and fix levels together comprise the program level or version of a program.

virtual printer. A view of a printer that refers only to the high-level data stream, such as ASCII or PostScript, that the printer understands. It does not include any information about how the printer hardware is attached to the host computer or the protocol used for transferring data to and from the printer.

volume. The basic unit of storage for a database, log file, or a storage pool. A volume can be an LVM logical volume, a standard file system file, a tape cartridge, or an optical platter. Each volume is identified by a unique volume identifier.

W

wildcard. Search characters that represent other letters, numbers, or special characters. In OnDemand, the percentage (%) and the underscore (_) symbols are wildcard characters.

window. A part of a display panel with visible boundaries in which information is presented.

workstation. A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can perform applications.

write access. In computer security, permission to write to an object.

writer. A JES function that processes print output.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbooks publication.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 636.

- ▶ *Content Manager OnDemand Backup, Recovery, and High Availability*, SG24-6444
- ▶ *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
- ▶ *IBM System Storage DR550 Setup and Implementation*, SG24-7091

Other resources

These publications are also relevant as further information sources:

- ▶ *IBM Content Manager OnDemand - User's Guide*, SC27-0836
- ▶ *IBM Content Manager OnDemand - Windows Client Customization Guide and Reference*, SC27-0837
- ▶ *IBM Content Manager OnDemand - Messages and Codes*, SC27-1379
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Administration Guide*, SC18-9237
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232

- ▶ *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*, GC18-9236
- ▶ *IBM Content Manager OnDemand - Web Enablement Kit Installation and Configuration Guide*, SC18-9231
- ▶ *IBM Content Manager OnDemand - Report Distribution: Installation, Use, and Reference*, SC18-9233
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - Configuration Guide*, GC27-1373
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - Administration Guide*, SC27-1374
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - Indexing Reference*, SC27-1375
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - Web Enablement Kit Installation and Configuration Guide*, SC27-1376
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - OnDemand Distribution Facility Installation Guide and Reference*, SC27-1377
- ▶ *IBM Content Manager OnDemand for z/OS and OS/390 - Introduction and Planning Guide*, GC27-1438
- ▶ *IBM Content Manager OnDemand for iSeries - Administration Guide*, SC41-5325
- ▶ *IBM Content Manager OnDemand for iSeries - Installation Guide*, SC41-5333
- ▶ *IBM Content Manager OnDemand iSeries Common Server - Planning and Installation Guide*, SC27-1158
- ▶ *IBM Content Manager OnDemand for iSeries Common Server - Administration Guide*, SC27-1161
- ▶ *IBM Content Manager OnDemand for iSeries Common Server - Indexing Reference*, SC27-1160
- ▶ *IBM Content Manager OnDemand for iSeries Common Server - Web Enablement Kit Installation and Configuration Guide*, SC27-1163
- ▶ *IBM Content Manager OnDemand for Multiplatforms Release Notes for Version 7.1.0.10* (comes with the Content Manager OnDemand for Multiplatforms software)
- ▶ *IBM DB2 UDB for z/OS and OS/390 - Administration Guide*, SC26-9931
- ▶ *Tivoli Storage Manager for Windows Administrator's Guide*, GC32-0782
- ▶ *Tivoli Storage Manager for AIX Administrator's Guide*, GC32-0768

- ▶ *Tivoli Storage Manager for Windows Quick Start*, GC32-0784
- ▶ *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- ▶ *z/OS MVS System Commands*, SA22-7627
- ▶ *DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426
- ▶ *OS/390 OpenEdit Command Reference*, SC28-1982
- ▶ *UNIX System Services Command Reference*, SC28-1892
- ▶ *PDF Reference, third edition, Adobe Portable Document Format Version 1.4*, Addison-Wesley, 2001, ISBN 0-201-75839-3
- ▶ *Adobe Type 1 Font Format*, Addison-Wesley, 1990, ISBN 0-201-57044-0
- ▶ The following publications are available with the Xenos offerings by Xenos Group Incorporated:
 - *Xenos d2e Platform User Guide*
 - *Xenos d2e Platform Scripting Reference*
 - *Xenos d2e Developer Studio User Guide*

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ IBM Content Manager OnDemand production information
<http://www.ibm.com/software/data/ondemand/>
- ▶ IBM Content Manager OnDemand Information Center
<http://publib.boulder.ibm.com/infocenter/cmod/v8r3m0>
- ▶ z/OS information
<http://www.ibm.com/servers/eserver/zseries/zos>
- ▶ OS/390 information
<http://www.ibm.com/servers/s390/os390>
- ▶ DB2 Universal Database for z/OS information
<http://www.ibm.com/software/db2zos/library.html>
- ▶ iSeries Information Center
<http://www.ibm.com/eserver/iseries/infocenter>
- ▶ iSeries Navigator information
<http://www.ibm.com/eserver/iseries/navigator/>

- ▶ IBM Printing Systems Division Web site for InfoPrint product information
<http://www.printers.ibm.com>
- ▶ Tivoli Storage Manager home page for Tivoli Storage Manager information
<http://www.tivoli.com/tsm>

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

Symbols

@SRV@_ARSSOCKET 99

A

About window

customization 511

display time customization 519

ACIF (Advanced Function Presentation and Indexing Facility) 10, 13, 31, 454

ACIF exits 337

index record exit 339

input record exit 337

output record exit 341

resource exit 342

ACIF user exit 337

Acrobat Distiller 198

Acrobat Reader 183

ACS

language 135

routine 135

translator 135

Add Report to OnDemand (ADDRPTOND) command 87, 156

ADDRPTOND (Add Report to OnDemand) command 87, 156

ad-hoc CD-ROM 18

mastering option 502

administration 21

guideline 53

Administration Client 600, 602

administrative client 13, 344

Adobe Acrobat 10, 13, 187

Adobe Toolkit 198

Adobe viewer 189

advanced application group

database information 156

storage management 155

Advanced Function Presentation (AFP) 2, 180

common indexing problem 463

conversion to XML through ODWEK 281

data 141, 180

fonts 180

image mapping 258, 264

presentation in PDF 251

printers 181

shaded images 260, 267

AFP (Advanced Function Presentation) 2, 180

AFP Conversion and Indexing Facility (ACIF) 10, 13, 31, 454

return code 16 338

AFP to PDF data conversion 251

AFP2HTML

conversion of AFP to HTML 253

with ODWEK 253

afp2html command 261

afp2html.ini file 255

AFP2PDF

conversion of AFP to PDF 261

example 305

ODWEK 262

transform with ARSLOAD 297

afp2pdf command 272

afp2pdf.ini file 263

afp2web command 261

AFP2WEB Services Offerings 249, 252

AFP2XML

AFP to XML conversion 272

example 281

module 272

AFPIndexer 275

AFPPARSER 284

AFPVIEWING parameter 253, 262

ag_internal_id 67

aggregate file, maximum size 149

aggregation

enables 148

process 149

Annotation parameter 37

annotation table 64, 68

annotations 2, 36

apka2e exit 338

APKACIF

exit on z/OS 386

exits written in COBOL 387

indexer program 387

application 5–8, 28, 30, 48, 50, 69, 84

defaults 482

definition 30
OnDemand 432
application group 5–8, 23, 25, 28, 48, 50, 69, 71, 75, 84, 432, 532, 543, 574–575
advanced settings 37
application added to existing group 45
Application ID field 443
data table 65–66
Date field 440
edit/view window 578
example 67
field alias table 64
Field Information tab 544
field table 64
Load Date field 441
name handling 446
permissions table 64
Right click 548
status field 481
storage management 123, 125, 152, 155
table 64, 434
table structure 68
window 589, 593
application ID 28–29
Application ID field in application group 443
application table 64
ArcCSxitApplGroup structure 364
ArcCSxitDocFields 365
ArcCSxitPermExit 369
ArcCSxitSecurityAction 374
ArcCSxitSecurityRC 374
architecture
ODWEK 200
OnDemand 162
Web server 174
archive copy group 107
archive documents 106
archive media 9, 12, 15
performance 169
archive storage 23, 25, 36, 116, 119, 123
Archive Storage Manager 105, 141–142
for iSeries 141
archive storage manager 9–10, 12, 14–15, 106
archive.mac 170
archived report 11
ARCHIVEPOOL 112
ARS_AUTOSTART_INSTANCE 90
ARS_DB_TABLESPACE 98
ARS_DB2_ARCHIVE_LOGPATH 81
ARS_DB2_DATABASE_PATH 81
ARS_DB2_PRIMARY_LOGPATH 81
ARS_NUM_DBSRVR parameter 167
ARS_OD_CFG 81, 98
ARS_TMP 84
ARS.CACHE 82, 88
z/OS 99
ARS.CFG 81, 88, 98, 115, 117
ars.cfg file 562
ARS.DBFS 83
ARS.INI 79, 85, 88, 92, 98, 100
z/OS 99
ARSADM 384
ARSAG 64, 69
ARSAG2FOL 64
ARSAGFLD 64
ARSAGFLDALIAS 64
ARSAGPERMS 64
ARSANN 64, 68
ARSAPP 64
ARSAPPUSR 64
arcsxit.h header file 361
ArsCSxitFaxOptions 364
ArsCSxitLoadExit 367
arsdate 11992 command 67
arsdate command 67
arsdb command 64, 83–84, 86
arsdb program 73–74, 100
ARSFOL 64
ARSFOLDPERMS 65
ARSFOLFLD 65
ARSFOLFLDUSR 65
ARSGROUP 65
ARSLOAD 280, 297, 387
AFP2PDF transform 297
installation verification 104
load table 69
modification for PDF indexing 196
modification for PDF indexing on z/OS 194
performance tuning 160
process 67, 75
running parallel jobs 167
Store OnDemand 494
VSAM data sets 140
with distributed object servers 163
Xenos transforms 279
arsload command 304, 596
ARSLOAD table 65
arslog 344

arslog file 344
 ARSLOG installation exit 349
 arsmaint command 12, 131
 ARSMINT expiration process 27, 124
 ARSMKDIR 95
 ARSNODE 65
 ARSPDF32.API 187
 ARSPRT 65
 arsprt file 354
 ARSPRTOPTS 65
 ARSPRTUSR 65
 ARSRES 68
 ARSSEG 69
 table structure 66
 ARSSET 65
 ARSSOCKD 84, 90
 ARSSOCKX 92
 ARSSYS 65
 ARSSYSCR 73, 102
 command 84
 arssyscr command 84, 86
 ARSUSRGRP 65
 ARSUSRGRPID 65
 arswww.ini file 171, 253–254, 262–263, 289
 ArsWWServlet 202
 ARSWWWSL.dll 206
 arsxenos.ini 290
 ARSXML 55
 add 56
 delete 56
 export 56
 update 56
 arsxml.cfg file 55
 ARSYSPIN 386
 ASCII 179
 asciinp exit 338
 asciinpe exit 339
 ATM font 180
 audit field 481
 authority 47–48, 52
 Automated Class Selection (ACS) routine 135

B

backup 115
 BACKUPPOOL 112
 banner 312, 319
 base 14 Type 1 fonts 186
 best practices 429
 for solution design 439
 bookmark 2
 BQT_DIST_ID 550–551
 BQT_DIST_NAME 550–551
 buffer pool 165
 buildICT module 272
 bulk document retrieval 209
 bundle 313, 321
 bundle content 562
 bundle entry 545, 547
 Large distributions 552, 554
 Bundle Query Table (BQT) 543

C

cache 26
 Cache Data 26, 123, 153
 expiration 132
 cache file system 168
 Cache Only 122
 storage node 122
 storage set 119
 Cache Only Library Server storage set 142
 cache parameters in ODWEK 171
 cache storage 15, 25, 36, 82, 134, 153, 432
 performance 169
 cache storage manager 9–12, 105–106
 cache threshold 124
 CacheDocs 173
 caching of documents 173
 CALLBACK 209, 213
 CALLBACK API 209
 cascading style sheet (CSS) 281, 293
 file sample 294
 CD image access from the CD-ROM 509
 CD-ROM
 access to CD image 509
 ad-hoc mastering 502
 Centera data protection 130
 centralization 47, 53
 CGI 173, 200, 202–203
 CLI trace for DB2 465
 Client Access Operations Navigator 144
 client logon problem diagnosis 467
 client node configuration 111
 client options file 113
 client program 3, 12
 client retrieval
 exit activation 373

preview exit 371
clnt_id parameter 374
coded data 132
collection 135, 138
Composite index xxix, 573, 578
 necessary fields 581
compressed data 132
compressed storage objects 125
concept
 application 5–6
 application group 5–7
 applications 7
 database manager 12
 document 4
 document indexing 8
 folder 6–7
 management programs 15
 report 4
 report indexing 9
 request manager 11
 server 9
 storage manager 12
configuration file 79
 ARS.CACHE 88
 ARS.CACHE, z/OS 99
 ARS.CFG 81, 88, 98
 ars.cfg 115, 117
 ARS.DBFS 83
 ARS.INI 79, 85, 88, 92, 98, 100
 ARS.INI, z/OS 99
 permission 82–83
 RS.CACHE 82
Content Manager OnDemand
 support 568
control file 257, 264, 485
control information 10
conversion
 load or transform 250
 of data streams 250
 program 10
crash, OnDemand server 469
Cross Reference Table
 reference value 546
Cross Reference Table (CRT) 532, 536
CSS (cascading style sheet) 281, 293

D

DAF (Document Audit Facility) 480

data
 attributes 27
 compression 183
 data type 28
 deletion 15
 indexing 11, 13–14
 indexing and loading 11, 13
 loading 6, 11, 13–14, 69
 loading programs 11
 migration 11
 migration from cache 125
 protection with Centera 130
 segmentation 24
 type
 filter 28
 index 28
 not in database 28
data control block (DCB) 195
data conversion 249
 integrated solutions with OnDemand 251
data field, segment 29
data stream conversion 250
data table 75
 structure 66
 application group data table 68
 ARSSEG 66
data type 562
database 12
 binding 83
 control information 10
 create OnDemand database 83
 creation and relationship on z/OS 73
 file system 168
 information 24
 locking 166
 logs 164
 maximum file handles 165
 objects 9
 problems 465
 relationship when loading data 69–70
 transaction logs 166
database manager 9–12
database structure 63
database view 432
Date field in application group 440
date range search tip 501
date, internal OnDemand format 67
DB2 control center 66
DB2 instance, creation of 79

DB2 library, export 101
DB2 parallelism 166
DB2 problem diagnosis 465
DB2 table 526
DB2INSTANCE 81, 98
db2setup 79
db2uexit.err 84
DBSIZE calculation 75
DCB (data control block) 195
debugging 207
 ODWEK 174
decentralization 47, 53
decentralized administrative model 49
decentralized system administration 49
decimal field, negative numbers 520
design of a winning solution 430
device class 108
device configuration 110
DFSMSdfp component 132
DFT_QUERYOPT 165
directories for OnDemand files 95
disconnected mode 18
disk 4, 106
disk pool storage group 145
Disk Storage Manager 142, 154, 156
DISKPOOL 112
Display Document Location 36
display list 284
distribution 313, 324
Distribution Control Table (DCT) 527, 533
distribution method 534, 539
Distribution Name (DIST-NAME) 533, 565
Distribution Request Table (DRT) 541
DLIST 284
dms script 286
doc file 569
document 4, 11, 26, 124, 154
 access 17
 access and distribution 17
 archival 106
 cache 173
 distribution possibility 18
 expiration type 132
 indexing 8
 manipulation script 283
 open password 270
 removal 15
 retrieval, bulk 209
 segmentation 31
transfer from OnDemand server to staging drive 503
Document Audit Facility (DAF) 480
 control file 485
document cache 545–546
document retrieval and delivery 310
download 11
 facility 10, 13
drive 169
dsm.opt 113
DSNEXIT 101
DSNLOAD 101

E

EBCDIC 179
EDTF command 88
e-mail 526–527
e-mail notification 538, 542
end user 48
example
 application 8
 application group 8
 application group ID 67
 folder 8
exit points 335
expiration
 attributes 133
 process 124
Expiration Type 26, 124, 154
 document 124, 132, 154
 load 124, 132
 segment 124, 131, 154
Extensible Stylesheet Language (XSL) 281
external link 96
external security exit 92
external storage 36

F

FAQ 454
fax options exit 336, 364
 activation 365
faxing documents 2
federated search 19
fenced UDF 79
Field Information 27
Field mapping table 64
file structure, OnDemand in UNIX 96
file systems in UNIX 167

filter 75
folder 5–7, 35, 48, 50, 84, 432, 446, 486
 creation 484
 modifications after migration 423
 primary folder 38
 query 28
 search 71–72
 search sequence 71
 secondary folder 38
folder field table 65
Folder List Filter 515
folder permission table 65
folder table 64
folder user field table 65
font 190
 listing in a PDF file 189
 map file 257
 mapping 180
font mapping table
 allocation 195
 creation 195
fontlib 181
FSA (functional subsystem application) 13
Full Report Browse 41
function 48
functional subsystem application (FSA) 13

G

generator 284
Generic Indexer 494
GIF 182
graphical indexer 190
group administration 47
group table 65

H

hang, OnDemand server 469
hard disk drive 111–112, 118, 134, 167, 170
 cache 123
header banner 312
header file arscsxit.h 361
HFS (hierarchical file system) 93, 95–96, 134
hierarchical file system (HFS) 93, 95–96, 134
hit list 37, 431, 514
 order 37
 single-selection 514
HP-UX 82, 84
HTTP server 173

HTTP Web server 200–201

I

I/O contention 168
 performance problem 168
IBM 3995 Optical jukebox 169
IBM AFP2WEB Services Offerings 252
IBM System Storage Archive Manager 106, 126
IBM WebSphere Application Server 173, 225
image 132
image data 182
image map file 257–258, 265
index 2, 75
 exit program 405
 fields 2
 record exit 339
Index field
 Key number 549
index field 532
index value 532, 544, 596
indexer parameter 33
indexing 6, 432
 composed AFP files 275
 data 8, 11, 13–14
 exits 336
 issue 454
 issues with PDF 188
 methods 8
 document indexing 8
 report indexing 9
 PDF 185
 PDF on z/OS 194
 problem diagnosis 463
Indexing parameter 13–14, 160
indexing program 6, 10, 13–14
 ACIF 10, 13
 OnDemand Generic Indexer 10
 OnDemand PDF Indexer 10
 OS/400 Indexer 10, 13
idxexit parameter 339
input record exit 337
 apka2e 338
 asciinp 338
 asciinpe 339
install library 95
invoice status 487
iSeries
 multiple instances 86

OnDemand 17
security exit 373
iSeries Common Server
analysis and planning 404
environment setup 400
migration 399
migration tool 400
ongoing use 427
Spool File Archive environment 402
iSeries Job Scheduler 427

J

Java API 201, 203
samples 206
Java Development Kit 206
JCL 13, 93, 196
JES
functional subsystem application 13
spool data 386
spool files 13
JES Spool Capture facility 386
Job Name 533–534
Job Script 284
program syntax 305
Job Supervisor program 304
JPEG 182

L

LAN (local area network) 163
large object 31
LCS (Library Control System) 133
Library Control System (LCS) 133
library issues 169
library server 3, 9–10, 78, 86, 162
functions 9
storage management 118
license 110
Life of Data and Indexes 26, 118–119, 124, 151, 153
line data 179
background, customization 517
load 6, 23, 26, 124, 154
image data 182
postprocessor 30
preprocessor 30
problems 178
single 25
VSAM data sets 141

load and indexing file system 168
Load Data 122, 137
load data server 86
Load Date field in application group 441
load exit 336, 366
activation 367
Load expiration type 132
load ID 69
Load Information 30
load issue 463
Load report 316
load table 65
ARSLOAD 69
Load Type 120, 136
document 119
fixed 120
local 120
segment 119
load-based schedule 330
loading
issue 454
problem diagnosis 463
loads per database table 25
local area network (LAN) 163
local printing 354
lock escalation 166
locking 166
LOCKLIST 166
log files 83
system log 84
logon 122
Lotus WordPro 11
LPAR 101

M

maintenance 24
management class 107, 134
management programs 11, 15
manifest 323
MAXFIOP 165
Maximum Hits 37
Maximum Rows field 75
maximum rows value 24
maximum size, aggregate file 149
message code 31 346
message logging 164
message number 87 299, 348
message number 88 299, 347

Message of the day 522
Migrate Data from Cache 156
migration 84
 analysis and planning 404
 iSeries Common Server 399
 migration tool 410
 modifications to folders after migration 423
 policy 142, 149, 152
 tool 400
 with the tool 420
 without the tool 410
mount retention 170
mountable file system 93
mountretention setting in Tivoli Storage Manager 170
Multiplatforms, OnDemand 16
multiple instances 77
 adding instance to ARS.INI 99
 connecting 85
 creation of tables 100
 creation on z/OS 103
 definition on iSeries 86
 definition on UNIX 78
 definition on Windows NT 86
 definition on z/OS 97
 overview on z/OS 92
 port number 80, 85, 88
 starting new instance 102
 z/OS 91
MVS 93, 95–96
MVS download server 86
MVS Dynamic Exit Facility 349

N

Named Query report 316
named query table 65
negative numbers in decimal fields 520
network performance 168
node table 65
note 37
Note icon 37
Note Search 36

O

OAM (object access method) 12, 139
OAM Storage Management Component 133
OAM Thread Isolation Support 133
OAMDASD 134

OAMOPTIC 134
OAMTAPE 134
object access method (OAM) 12, 139
 API functions 133
 collection 135
 components 133
 Library Control System (LCS) 133
 OAM Storage Management Component (OSMC) 133
 Object Storage and Retrieval (OSR) 133
 for z/OS 132
 object expiration 139
object maximum size 139
object naming conventions 138
object owner model 49, 51
 administrator roles 52
 implementation 52
object server 3, 9–10, 106, 162
 functions 9
object size 125, 155
object storage 133
object storage and retrieval 133
object type model 49
 administrator roles 50
 implementation 49
OD Delete application 497
OD File System Monitor application 498
OD Store application 497
OD Update application 497
ODWEK
 AFP2HTML 253
 AFP2PDF 262
 APIs 202
 architecture 200
 cache 170
 performance 173
 sizing process 171
 caching documents 173
 configuration 170, 289
 configuration of Xenos transforms 289
 conversion of AFP to XML 281
 debug 174
 debugging 174
 logging 174
 problem diagnosis 468
 servlet deployment 225
ODWEK (OnDemand Web Enablement Kit) 159, 200, 279, 461
OLTP (online transaction processing) 165

OMVS 101
 shell 92

OnDemand 2, 21, 431, 525–526
 administrative client trace parameters 477

 AFP2WEB Services Offerings 252

 application 432

 application group 432

 architecture 162

 batch commands in z/OS 499

 bulletins 524

 client issue 460

 client logon problem diagnosis 467

 concepts 4

 configuration 163

 configuration for Tivoli Storage Manager archive management 115

 configurator 116

 database 164

 document access and distribution policies 17

 file structure in UNIX 96

 folder 432

 Generic Indexer 494

 IBM System Storage Archive Manager 126

 instance 78

 integrated solutions for data conversion 251

 maintenance 458

 management programs 11

 PDF and client 187

 PDF Indexer 10

 performance 159

 Production Data Distribution (PDD) 511

 server 9

 solution design 430

 splash screen, display time customization 519

 startup problem 459

 storage 432

 storage management 118

 supported environments 15

 system flow 22

 system management 166

 temporary space file system 168

 trace facility 474

 tuning to enhance performance 162

 Xenos 278

OnDemand Administration GUI 139

OnDemand administrator 556, 574

OnDemand Administrator Client 151

OnDemand configuration
 distributed 10

 single 9

OnDemand Distribution Facility
 new feature Reference Field Caching Option 545

OnDemand Distribution Facility (ODF) 526–527

OnDemand for iSeries 17

OnDemand for Multiplatforms 16

OnDemand for UNIX, Tivoli Storage Manager configuration 117

OnDemand for Windows Tivoli Storage Manager configuration 116

OnDemand for z/OS 16

OnDemand Generic Indexer 10

OnDemand server 527, 544, 576, 578, 600
 document transfer to staging drive 503

 hang or crash 469

 mode 18

 query performance 550

OnDemand server components 10
 cache storage manager 10

 conversion program 10

 data loading program 11

 database manager 10

 download facility 10

 indexing programs 10

 request manager 10

OnDemand Toolbox 496, 525, 568

OnDemand user 556

OnDemand V7 532, 536

OnDemand Web Enablement Kit (ODWEK) 159, 200, 279, 461

OnDemand Windows Client 555–556

OnDemand XML Batch Administration 53

ondemand.xsd schema file 55

online transaction processing (OLTP) 165

open database files 165

operating system performance 168

optical 4, 9, 12, 106

optical storage 432

optical storage group 146

optimization class 165

option file 269

OS/400 auxiliary storage pool 145

OS/400 Indexer 10, 13

OSMC 133

OSR 133

OSREQ assembler macro 133

OTIS 133

outexit parameter 341

output record exit 341
owner password, master password 271

P

parser 284
partitioned data set (PDS) 195
PDD (Production Data Distribution) 511
PDF 10, 13, 251
 architecture 178
 data 177
 data indexing 188
 definition 186
 document access 449
 document indexing 451
 file listing fonts 189
 file size, loading 188
 graphical indexer 192
 indexing 185
 on z/OS 194
 indexing issues 188
PDF (Portable Document Format) 177, 186
PDF Indexer 602
PDF indexer 190
 limitations 198
 maximum file size 178
 testing in z/OS 500
PDS (partitioned data set) 195
performance 24, 31, 36, 109, 132, 135, 207
 data compression 183
 data loading 160
 database 164
 debugging ODWEK 174
 enhancements 163
 I/O contention 168
 image data 182
 issues based on data type 177
 line data 179
 memory 165
 mount retention 170
 network 168
 ODWEK cache 170, 173
 ODWEK logging 174
 ODWEK with multiple user ID access 175
 ODWEK with single user ID access 176
 OnDemand 159
 open database files 165
 operating system 168
 organize file systems on UNIX 167

PDF data 177
physical media and library issues 169
problem diagnosis 468
query optimization 165
running large load jobs 167
running parallel load jobs 167
servlets versus CGI 173
storage management 169
system logging 164
text search 40
transaction data 179
troubleshooting reference 161
tuning when is necessary 160
performance configuration 109
permission 104
 password 271
 UNIX file permission bits 94
 UNIX file permissions 94
 updating for various users 483
permission exit 336, 367
 activation 371
physical media
 issues 169
policy domain 107, 111
port number 80, 85, 88, 99, 102
Portable Document Format (PDF) 177, 186
PostScript language 186
primary allocation 75
primary folder 38
primary logs file system 168
primary storage group 151
primary storage node 137
print exit for Multiplatforms 354
Print Processor
 entry 542
 Table 543
Print Query Table (PQT) 543
Print Services Facility (PSF) 10
 for OS/390 13
printer options table 65
printer table 65
printer user table 65
problem diagnosis categories 454
Production Data Distribution (PDD) 511
PSF (Print Services Facility) 10

Q

query optimization 165

R

RACF 92, 385
Redbooks Web site 636
 Contact us xxvi
Reference field 527, 544
 index value 545
registry, modification of client behavior 513
Related Documents
 configuration 491
 feature 489
report 4–6, 9, 312
 administration 22
 administrator 48–50, 52
 distribution 18, 307
 download 13
 indexing 9
 trigger 192
 type 316
report data 554
Report Distribution 307
 addition of a report 316
 definition of distribution 311
 document delivery 311
 document retrieval and delivery 310
 hints and tips 330
 necessary documents 310
 parameters 328
 who receives the documents 310
Report field definition 193
report ID 532, 535
Report Name
 Distribution definition 542
report name 537, 542
Report Wizard 13, 43, 190, 194, 577
reprinting documents 2
Request manager 9–12
resexit parameter 342
resource 2
 exit 342
 file 11
 group 181
 table 68
resources table 65
restore 115
restype parameter 342
retention period 118–119
RETINIT 127
return code 16 338
REXX 286

S

SAF 378
SAF resource classes 384
same time 552
sample code
 capture failed logon 346
 logon and search 207
 notify when load completes 347
 search with CALLBACK 209
 send e-mail when load fails 346
 updating document index 214
sample registry import file 492
SAVCHGOBJ (Save Changed Objects) command 157
Save Changed Objects (SAVCHGOBJ) command 157
schedule 313, 320
schema file 55
search criteria 71
search sequence 72
search sequence from folder 71
search with CALLBACK 209
secondary allocation 75
Secondary Folder 38
secondary logs file system 168
security 270
 external security exit 92
 OS/400 89
security exit 336
 activating 384
 iSeries 373
 modules 377
 on z/OS 376
segment 26, 29, 124, 154
 expiration type 131
 fields 434
 table 65–66, 69
segmentation of index data 24
separator banner 312
server 9
 initialization 110
 library server 9
 object server 9
 print facility 11
 print manager 9
 printing 354
 programs 3
services offering 570
servlet 173, 200, 203

custom 202
shortcut 2
Single Coded Character Set ID 87
single load 25
single-selection hit list 514
smitty 79
SMS
 management class 134
 storage class 134
 storage group 134
 terminology 133–134
solution design 429
 bad example 436
 best practices 439
 case study 436
 good example 437
 three-step approach 432
solution, winning design 430
SPACEMGPOOL 112
splash screen, display time customization 519
Spool File Archive environment 402
SPUFI 66
SQL query 550, 555
SQL report 316
SRV_OD_CFG 99
SRV_SM_CFG 99
SRVR_FLAGS_SECURITY_EXIT 89
SRVR_INSTANCE 80, 99
SRVR_INSTANCE_OWNER 80, 99
SRVR_SM_CFG 82
Start Disk Storage Management (STRDSMOND)
 command 12, 154
 status field 481
 in application group 481
storage 4, 432
 archive media 9, 12
 cache 15, 82
 disk 4, 106
 long term 122
 optical 4, 9, 12, 106
 tape 4, 9, 12, 106
Storage Archive Manager 126
storage class 134
storage classes 134
 OAMDASD 134
 OAMOPTIC 134
 OAMTAP 134
storage devices 3
storage group 134
storage level 149
storage management 25, 105, 118
 advanced application group 155
 application group 123, 125, 152
 expiration 139
 external cache exit 336, 393
 activation 394
 object size 125
 performance 169
storage manager 11–12
 archive storage manager 9–10, 12, 14–15, 106
 cache storage manager 9–12, 106
storage node 119, 121
storage node name 121
storage policy 107
storage pool 111–112
storage pool hierarchy 113
storage pools and volumes 108
storage set 23, 26, 48, 119, 121, 153, 574–575
 cache only 122
 definition 136
 load data 122
 load type 120
 logon 122
 storage node 121
storage set definition 23
storage set table 65
Store OnDemand 493
 what it does 495
 why it is needed 494
STRDSMOND (Start Disk Storage Management)
 command 12, 154
STRMONOND 87, 156
STRTCPSVR 90
Sun Solaris 82, 84
symbolic link 96
SYSOUT 386
system administration 47, 344
 decentralized 49
system administrator 47–50, 52
system control tables 64
 ARSAG 64, 69
 ARSAG2FOL 64
 ARSAGFLD 64
 ARSAGFLD ALIAS 64
 ARSAGPERMS 64
 ARSANN 64, 68
 ARSAPP 64
 ARSAPPUSR 64

ARSFOL 64
ARSFOLDPERMS 65
ARSFOLFLD 65
ARSFOLFLDUSR 65
ARSGROUP 65
ARSLOAD 65, 69
ARSNAMEQ 65
ARSNODE 65
ARSPRT 65
ARSPRTOPTS 65
ARSPRTUSR 65
ARSRES 65, 68
ARSSEG 65, 69
ARSSET 65
ARSSYS 65
ARSUSER 65
ARSUSRGRP 65
ARSUSRGRPID 65
system design 47
system log 12, 73, 84, 87, 91
 initializing 102
system log exit 344
 for Multiplatforms 344
 samples 345
 z/OS 349
system log message 80 488–489
system logging 11, 164
system management 166
system parameters 164
system parameters table 65
system printer 48
system size 47
system tables 73

T

T date search option 501
T format string 501
table builder 432
tablespace 73–75, 83, 98
tablespace creation exit 336, 394
 activation 397
tape 4, 9, 12, 106
tape storage 432
TCP/IP 3–4
text search 38
thumbnail 2
TIFF 11, 13, 182–183
Tivoli Storage Manager 12, 108–109, 115

archive copy group 107
backup 113
Client node 107
client node 121
client node configuration 111, 113
components 117
configuration 108
configuration of OnDemand for archive management
 115
configuration of OnDemand for UNIX 117
configuration of OnDemand for Windows 116
configuration verification 113
device class 108
device configuration 110
drive 108
for Multiplatforms 106
installation 108
installation verification 113
library 108
license 110
licenses 108
logon 122
management class 107
options files 117
overview 106
performance configuration 109
policy domain 107, 111, 119
problem diagnosis 467
storage policy 107
storage pool 111–112
storage pools and volumes 108
wizard 108

TMerge 284
trace facility 474
trace parameters in OnDemand administrative client
477
trailer banner 312
trailing negative sign 521
transaction 9
 data 179
 indexing 179
 log 4, 166
 number 12
 report 422
 value 5
transform profile file 257
trigger 191
troubleshooting 169, 453–454

arsxml 60
ODWEK and Xenos 296
performance 161
TT font 180
Type 1 font 186

U

Unicode format of index parameter 33
UNIX
 file permissions 94
 multiple instances 78
UNIX System Services 92–93
 file system 74
UPDATE API 214
user access 6, 47, 175
 ODWEK 175–176
user administration 47
user administrator 48–50, 52
user exit 12, 83, 336
 ACIF 337
 customized functions 360
 debugging 342
 fax options exit 336
 header file arscxit.h 361
 indexing exits 336
 load exit 336
 permissions exit 336
 print exit for Multiplatforms 354
 security exit 336, 373
 security exit on z/OS 376
 storage management external cache 393
 storage management external cache exit 336
 system log 12
 system log exit for Multiplatforms 344
 tablespace creation exit 336, 394
user group ID table 65
user ID 175
user logical views table 64
User Output Table (UOT) 543
user password 270
user permissions 35, 104
user table 65
user type 48
users in group table 65

V

version support 444
view of the database 432

Virtual Storage Access Method (VSAM) 12, 140
 data sets 141
 storing data to data sets 140
VSAM (Virtual Storage Access Method) 12, 140

W

WAN (wide area network) 163
Web Enablement Kit 304
Web interface 572
Web server architecture 174
WebSphere Application Server 173, 225
WebSphere Information Integrator Content Edition,
federated search 19
wide area network (WAN) 163
Windows Client 598
Windows NT multiple instances 86
winning solution design 430
Work with Active Jobs (WRKACTJOB) command
90
workload balance 174
WORM (Write Once Read Many) 126
Write Once Read Many (WORM) 126
WRKACTJOB (Work with Active Jobs) command
90

X

Xenos
 and OnDemand 278
 d2e from Xenos 249
 document manipulation script 286
 job parameter file 283
 parameter file 284
Xenos transforms 280
 available dynamic transforms via ODWEK 279
 available transforms at load time 279
 ODWEK configuration 289
Xerces2 Java Parser 54
Xerox metacode 251
XML Batch Administration program 53
XML Batch Administration program (XML batch pro-
gram) 53
XML document viewing 292
XSL (Extensible Stylesheet Language) 281

Z

z/OS
 APKACIF exit 386

batch OnDemand commands 499
database creation and relationship 73
multiple instances 91
object access method (OAM) 132
testing PDF indexer 500

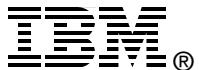
IBM



Redbooks

Content Manager OnDemand Guide

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



Content Manager OnDemand Guide



Administration, database structure, and multiple instances

This IBM Redbooks publication covers a variety of topics relating to the practical application of Content Manager OnDemand (simply referred to as "OnDemand") for Multiplatforms Version 8.3 (also known as Version 7.1.2.5), z/OS Version 7.1, and IBM eServer iSeries Common Server Version 5.3 of the OnDemand product. Where necessary, separate sections are included to cover variations between the different platforms.

Storage management, PDF indexing, and data conversion

This IBM Redbooks publication provides helpful, practical advice, hints, and tips for those involved in the design, installation, configuration, system administration, and tuning of an OnDemand system. It covers key areas that are either not well known to the OnDemand community or are misunderstood.

Exits, iSeries CS migration, best practices, and many more

We reviewed all aspects of the OnDemand topics. Among these topics, which we present in this IBM Redbooks publication, are administration, database structure, multiple instances, storage management, performance, PDF indexing, OnDemand Web Enablement Kit, data conversion, report distribution, exits, and iSeries Common Server migration.

Because a number of other sources are available that address various subjects on different platforms, this IBM Redbooks publication is not intended as a comprehensive guide for OnDemand. We step beyond the existing OnDemand documentation to provide insight into the issues that might be encountered in the setup and use of OnDemand.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:
ibm.com/redbooks**