

*Content Manager OnDemand
for Multiplatforms Version 7.1
Release Notes*

August 28, 2003



Content Manager OnDemand for Multiplatforms Version 7.1 Release Notes



Content Manager OnDemand for Multiplatforms Version 7.1 Release Notes

Note

Before using this information and the product it supports, read the information in "Notices" on page 235.

Sixteenth Edition (September 2003)

This edition applies to IBM Content Manager OnDemand for Multiplatforms, Version 7 Release 1 and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2001, 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About the Release Notes	vii
Change history	vii

Part 1. General release notes 1

Chapter 1. ACIF indexing 3

Attributes of the input file	3
Concatenating resources to an AFP file	3
CONVERT parameter	4
EXTENSIONS parameter	4
FIELD parameter	4
Constant field	4
DEFAULT keyword	4
MASK keyword	5
INDEX parameter	6
INDEXOBJ parameter	6
MCF2REF parameter	7
Messages	7
Page-level indexes	7
PRMODE parameter	7
RESTYPE parameter	8
Running ACIF with inline resources	8
TRIGGER parameter	8
Group triggers	8
Understanding how ACIF processes fully composed AFP files	9
Using ACIF	10
Writing inline resources to the output file	10

Chapter 2. Adobe software 11

Chapter 3. AFP information 13

AFP to line data conversion	13
AFP viewing on Windows XP	13

Chapter 4. AIX information 15

AIX 4.3.3 or later	15
AIX Fix Pack 9	15

Chapter 5. ARS.CFG 17

Chapter 6. ARSJESD user exit. 19

ARSJESD processing	19
Sample user exit program	19

Chapter 7. ARSLOG user exit 23

Chapter 8. ARSSOCKD stop command 25

Chapter 9. Command reference 27

General information	27
ARS_ADSM	28
ARSADM	28

Deleting applications, application groups, and folders	28
SET function	29
USER function	30
ARSADMIN	30
ARSDATE	30
ARSDB	31
Specifying backup devices	31
Backing up to TSM storage	31
ARSDOC	32
ADD function	32
-f parameter	32
GET function	32
Messages	32
QUERY function	33
-S parameter	33
UPDATE function	33
-x parameter	34
-X parameter	34
ARSLOAD	35
ARSMOINT	35
ARSTBLS	35

Chapter 10. DB2 UDB Version 8.1 information 37

Chapter 11. DB2 UDB Version 7 information 39

UNIX servers	39
Windows servers	39

Chapter 12. Generic indexer 41

CODEPAGE: parameter	41
GROUP_FILENAME: parameter	41
Parameter syntax	41

Chapter 13. Hardware and software requirements 43

Chapter 14. Installing on UNIX 45

Creating a user for the OnDemand instance owner	45
Setting permissions	46
Configuring the ARS.INI file	46
Configuring object servers	46

Chapter 15. Messages 49

Chapter 16. Migrating from previous versions of OnDemand. 51

Preparing databases for migration	51
Preparing archive storage for migration	52
Migrating from a previous version	52

Chapter 17. Naming rules. 57

Chapter 18. National language support 59

Character data conversion tables 59
Introduction 59
Obtaining source conversion tables 59
Modifying ICONV tables 60
Generating an updated ICONV table 61
Distributing an updated conversion table 61
ICONVDEF program reference 61
Language codes in UNIX 62
DBCS code pages 62
Mapping AFP fonts. 63
Traditional Chinese clients 63

Chapter 19. Oracle information 65

Support for Oracle 9i 65
Installation notes for UNIX servers 65
Field types and field sizes 66
Estimating database storage space 67
Estimating the size of rollback segments. 68
Tuning the database 69
Using the ARSDB program 70
Using the ARSMAINT program 70

Chapter 20. Password security 73

Chapter 21. PDF Indexer 75

General information 75
Specifying the location of Adobe fonts 75
TRACEDD parameter 77
Trace facility 77
Processing PDF input files with the graphical indexer 79

Chapter 22. Publications 83

Chapter 23. Publications CD-ROM 85

CD-ROM directory structure. 85
OnDemand publications 86
Viewing the HTML books 86
Printing the PDF books 87
Ordering the printed books 87

Chapter 24. Remote library server 89

Chapter 25. Retrieval preview user exit 91

Chapter 26. Support 93

Chapter 27. System control tables 95

List of tables 95
Application group table (ARSAG) 96
Field mapping table (ARSAG2FOL) 97
Application group field table (ARSAGFLD). 98
Application group field alias table (ARSAGFLDALIAS) 99

Application group permissions table (ARSAGPERMS) 100
Annotation table (ARSANN) 101
Application table (ARSAPP) 102
User logical views table (ARSAPPUSR). 103
Folder table (ARSFOL) 104
Folder field table (ARSFOLFLD) 105
Folder user fields table (ARSFOLFLDUSR) 106
Folder permissions table (ARSFOLPERMS) 107
Group table (ARSGROUP) 108
Load table (ARSLOAD) 109
Named query table (ARSNAMEQ) 110
Node table (ARSNODE) 111
Printer table (ARSPRT) 112
Printer options table (ARSPRTOPTS) 113
Printer user table (ARSPRTUSR) 114
Resources table (ARSRES) 115
Segment table (ARSSEG) 116
Storage set table (ARSSET) 117
System parameters table (ARSSYS) 118
User table (ARSUSER) 119
Users in groups table (ARSUSRGRP) 120
User group ID table (ARSUSRGRPID) 121
Application group data table 122

Chapter 28. Table space file systems 123

Chapter 29. TSM information. 125

TSM Version 5.1 125
TSM Version 4.1 126
Deleting application groups 126

Chapter 30. Web Enablement Kit 127

AFP Web Viewer 127
ARSWWW.INI file. 127
AddFieldsToDocId 127
ShowSearchString 128
CGI API reference 128
Document Hit List. 128
Configuring ODWEK for DBCS languages. 129
Image Web Viewer 130
Internet Explorer 130
Java APIs. 131
Shared library 133

Chapter 31. Windows client customization 135

GetStoreDocInvalidFieldNum 136
SetResourceCacheMode 139
StoreDoc 141
RetrieveDoc 144
STORE_DOC 144
Modifying client behavior through the Registry 144

Chapter 32. Windows server information 147

Windows Server 2003. 147
ARSLOAD service failure 147
Database backup 148

Part 2. System administration	149
Chapter 33. Overview	151
Chapter 34. User types	153
Chapter 35. Authority	155
Chapter 36. Permissions	159
Users	159
Groups	159
Applications.	160
Application Groups	160
Folders	160
Storage sets	161
Printers	161
Chapter 37. System administration	163
Object Type model	163
Object Owner model	164
Chapter 38. Summary	167
Chapter 39. Helpful hints	169
Chapter 40. Local server setup	171
Part 3. Xenos transforms	175
Chapter 41. Understanding Xenos	177
Chapter 42. Xenos transforms	181
Convert data streams.	181
AFP documents to HTML	181
AFP data to PDF	182
AFP documents to XML	182
Metacode to AFP	182
Metacode documents to HTML	183
Metacode to Metacode	183
Metacode to PDF	183
Metacode documents to XML	184
PCL to PDF	184
Index documents	184
Indexing with data values	185
Collect resources	185
Summary.	186
Chapter 43. Loading data	189

Chapter 44. Scenarios for using Xenos	191
Viewing and printing with the Windows client	192
Viewing and printing with the Web Enablement Kit	194
Chapter 45. How to specify parameters to the Xenos transforms	197
Processing sample data	197
Specifying parameters to OnDemand	198
Chapter 46. JS program reference	203
Purpose	203
Syntax.	203
Description	203
Parameters	204
Examples.	205
Generating the locations of text strings on a page	207
Indexing and converting input data	212
Chapter 47. Configuring ODWEK	221
Configuring the ARSWWW.INI file	221
[XENOS]	221
Specifying the AFP transform option	222
Specifying the Metacode transform option.	223
Configuring the ARSXENOS.INI file.	223
Specifying the ARSXENOS.INI file	224
Viewing converted documents.	225
Example of a parameter file	225
Example of a script file	227
Changes to the Retrieve Document API.	229
Accessibility features	231
Keyboard input and navigation	231
Keyboard input.	231
Keyboard focus.	231
Features for accessible display.	231
High-contrast mode	231
Font settings.	231
Non-dependence on color	231
Alternative alert cues.	232
Compatibility with assistive technologies	232
Accessible documentation	232
User exits	233
Notices	235
Copyright statement	235
Trademarks	235

About the Release Notes

The information about OnDemand contained in this Release Notes file was not available at the time that the OnDemand product publications were produced. Please make a note of these items, which supersede the information in the OnDemand product publications. For the latest information about OnDemand, see our Web site at:

<http://www.ibm.com/software/data/ondemand/mp>

Change history

1. Version 7.1.0.0 First release March 30, 2001.
2. Version 7.1.0.1 Second Edition April 20, 2001. Added or updated the following information:
 - ACIF indexing. Added the “INDEXOBJ parameter” on page 6.
 - Chapter 2, “Adobe software,” on page 11. Added a note about the ARSPDF32.API file. Customers should purchase Adobe Acrobat or Adobe Acrobat Business Tools to view PDF documents from the client. Added a note about processing PDF input files with the administrative client.
 - Chapter 13, “Hardware and software requirements,” on page 43. Updated the requirements for the CICS/ESA[®] client. Added a note about the OnDemand PDF Indexer feature. Added a note about Adobe Acrobat. Added a note about processing PDF input files with the administrative client.
 - Chapter 21, “PDF Indexer,” on page 75. Added a section on how to use the graphical indexer to generate indexing information for PDF input files.
 - Chapter 26, “Support,” on page 93. Added the component IDs for the clients and the server.
 - Xenos transform. Replaced the sample script file in “Indexing and converting input data” on page 212.
3. Version 7.1.0.2 Third Edition May 18, 2001. Added or updated the following information:
 - ACIF indexing. Added definition of a group to the TRIGGER parameter (see “TRIGGER parameter” on page 8).
 - Added a note about using the -I parameter to specify the name of the OnDemand instance to process with the ARSDB and ARSLOAD programs (see Chapter 9, “Command reference,” on page 27).
 - Chapter 12, “Generic indexer,” on page 41. Added a section with corrections for the generic indexer.
 - Chapter 14, “Installing on UNIX,” on page 45. Added a section on configuring OnDemand server to run under a user other than the root user.
 - Chapter 21, “PDF Indexer,” on page 75. Added a section to update the system limitations and NLS considerations for the PDF indexer.
 - Chapter 30, “Web Enablement Kit,” on page 127. Added browser requirements for the Web Enablement Kit (ODWEK). Updated server requirements. Documentation for the Java[™] APIs is now available. Updates to the No HTML output function. Added the ALLOBJECTS directive to the AFP2PDF.INI file.

4. Version 7.1.0.3 Fourth Edition July 13, 2001. Added or updated the following information:
 - ACIF indexing. Added a correction to the documentation on the attributes of the input file. (See “Attributes of the input file” on page 3.)
 - ACIF indexing. Added an important note about indexing fully composed AFP™ files (input files that contain BNG and TLE Structured Fields). See “Understanding how ACIF processes fully composed AFP files” on page 9.
 - Added an important note about backing up the TSM database with the ARS_ADSM program. (See “ARS_ADSM” on page 28.)
 - Added a requirement for AIX® 4.3.3 or later servers to install the C Set ++® Runtime for AIX 4.3 before installing or upgrading to OnDemand Version 7.1 (see Chapter 13, “Hardware and software requirements,” on page 43).
 - Added corrections to the ARSWWW.INI file. See Chapter 30, “Web Enablement Kit,” on page 127.
5. Version 7.1.0.4 Fifth Edition September 21, 2001. Added or updated the following information:
 - Updated the -E parameter for the ARSADM program. (See “ARSADM” on page 28.)
 - The ARSADMIN program now supports the ICONV function. (See “ARSADMIN” on page 30.)
 - Added the -x and -X parameters and updated the -S parameter for the ARSDOC program. (See “ARSDOC” on page 32.)
 - Correction to the -X parameter for the ARSLOAD program. (See “ARSLOAD” on page 35.)
 - Added corrections to the ARSWWW.INI file. See Chapter 30, “Web Enablement Kit,” on page 127.
 - Correction to the locale for the CHT language code on AIX and HP-UX. (See Chapter 18, “National language support,” on page 59.)
 - Added a correction to the documentation on the Logon method. (See Chapter 31, “Windows client customization,” on page 135.)
 - Update to the FIELD parameter for the PDF indexer. (See Chapter 21, “PDF Indexer,” on page 75.)
 - Update to the MASK option of the FIELD parameter for the PDF indexer. (See Chapter 21, “PDF Indexer,” on page 75.)
6. Version 7.1.0.5 Sixth Edition December 14, 2001. Added or updated the following information:
 - ACIF indexing. The OBJCON keyword is valid with the RESTYPE parameter. (See “RESTYPE parameter” on page 8.) ACIF can now generate page-level indexes without converting the input line data to AFP. (See “CONVERT parameter” on page 4, “INDEX parameter” on page 6, and “Page-level indexes” on page 7.) Added a note about constant fields and fields that are based on a floating trigger. (See “FIELD parameter” on page 4.)
 - Chapter 30, “Web Enablement Kit,” on page 127. Added the SERVERPRINTERS directive to the Default Browser section of the ARSWWW.INI file. Added the USEEXECUTABLE directive to the AFP2PDF section of the ARSWWW.INI file. Added installation notes on the WebSphere® Application Server. Added the ALLOBJECTS directive to the AFP2HTML.INI file.

- The SET function of the ARSADM program now supports storage nodes on an OnDemand for z/OS™ and OS/390® V7.1 object server. (See “ARSADM” on page 28.)
 - Added the capability to define a remote library server using the Windows® server configurator program. (See Chapter 24, “Remote library server,” on page 89.)
 - Correction to methods that contain an error in the way that String variables are declared in the Visual Basic example that is provided with the method. (See Chapter 31, “Windows client customization,” on page 135.)
7. Version 7.1.0.6 Seventh Edition February 28, 2002. Added or updated the following information:
- ACIF indexing. The MASK keyword of the FIELD parameter now supports a field that is based on a floating trigger. (See “FIELD parameter” on page 4.)
 - Chapter 2, “Adobe software,” on page 11. Added Acrobat Approval to the list of Adobe products that can be used to view PDF documents from the Windows client and to work with PDF input files from the administrative client.
 - Added a note about a potential problem with AIX Fix Pack 9 and the ARSSOCKD and ARSLOAD programs. (See Chapter 4, “AIX information,” on page 15.)
 - The ARSADM program now supports the deletion of applications, application groups, and folders. (See “ARSADM” on page 28.)
 - For the ARSMAINT and ARSTBLSP programs, added a note about using the -I parameter to specify the name of the OnDemand instance to process (see Chapter 9, “Command reference,” on page 27).
 - Chapter 13, “Hardware and software requirements,” on page 43. Added a note about Acrobat Approval. Added a note about a potential problem with AIX Fix Pack 9 and the ARSSOCKD and ARSLOAD programs.
 - Added a note about manually deleting application group data from TSM, when the owner of the OnDemand instance is not root. (See Chapter 29, “TSM information,” on page 125.)
 - Chapter 30, “Web Enablement Kit,” on page 127. Beginning with version 7105, IBM® now provides the IBM OnDemand AFP Web Viewer and the IBM OnDemand Image Web Viewer. Added a note about accessing an OD/390 V2.1 server with ODWEK 7106. Added the TempDir directive to the CONFIGURATION section. Added the ReportServerTimeout directive to the SECURITY section. Correction to the AddFieldsToDocid directive.
 - Correction to the PRINTDOC method and update to the Document Audit Facility. (See Chapter 31, “Windows client customization,” on page 135.)
8. Version 7.1.0.7 Eighth Edition May 3, 2002. Added or updated the following information:
- ACIF indexing. A new ACIF messages reference is available (see “Messages” on page 7). Correction to the “MCF2REF parameter” on page 7. Update to “Understanding how ACIF processes fully composed AFP files” on page 9.
 - Added a note about omitting the -g or -G parameter when running the ARSDOC program (see Chapter 9, “Command reference,” on page 27).
 - Updated the list of installable components for DB2 Universal Database™ (DB2® UDB) V7 on UNIX® library servers. See Chapter 11, “DB2 UDB Version 7 information,” on page 39.

- Added a version requirement for using Lotus Notes[®] to send documents from the OnDemand client. Added a version requirement for using the Java API that is provided with the Web Enablement Kit. See Chapter 13, “Hardware and software requirements,” on page 43.
 - Added a section on mapping AFP fonts for multi-byte languages. See Chapter 18, “National language support,” on page 59.
 - Chapter 21, “PDF Indexer,” on page 75. Updated the system limitations on input file size, number of pages, and available memory.
 - Add a note about the new version of *IBM Content Manager OnDemand: Messages and Codes*, SC27-1379-01. See Chapter 22, “Publications,” on page 83
 - Added the ENCRYPTCOOKIES and ENCRYPTURL parameters to the ARSWWW.INI file. Updated the CACHEDIR and TEMPDIR parameters in the ARSWWW.INI file. Corrected an error in the Retrieve document API Reference. Added a version requirement for using the Java API that is provided with the Web Enablement Kit. See Chapter 30, “Web Enablement Kit,” on page 127.
 - Added an important note on resolving a problem that may cause the ARSLOAD service to fail to start on Windows servers. See Chapter 32, “Windows server information,” on page 147.
 - Added a section on customizing the administrative client to handle exporting definitions between local and non-local servers. See Chapter 40, “Local server setup,” on page 171.
9. 7.1.0.8 Ninth Edition July 2, 2002. Added or updated the following information:
- Chapter 2, “Adobe software,” on page 11. Added a note about ODWEK, Internet Explorer, and Acrobat.
 - Added an export option to the solution for the potential problem with AIX Fix Pack 9 and the ARSSOCKD and ARSLOAD programs. (See Chapter 4, “AIX information,” on page 15.)
 - Added a note about ODWEK, Internet Explorer Acrobat. Added a version requirement for using the Java servlet that is provided with the Web Enablement Kit. Added a note about support for the TSM Version 5.1 server and client API with OnDemand. See Chapter 13, “Hardware and software requirements,” on page 43.
 - Add a note about updated versions of *IBM Content Manager OnDemand: User’s Guide*, SC27-0836-01 and *IBM Content Manager OnDemand: Windows Client Customization Guide*, SC27-0837-01. See Chapter 22, “Publications,” on page 83
 - Added information about support for TSM Version 5.1. (See Chapter 29, “TSM information,” on page 125.)
 - Added corrections and clarifications to the information about the CACHEDIR and TEMPDIR parameters in the ARSWWW.INI file. Added the APPLCACHEDIR parameter to the ARSWWW.INI file. Added a version requirement for using the Java servlet that is provided with the Web Enablement Kit. Added a note about ODWEK, Internet Explorer and Acrobat. See Chapter 30, “Web Enablement Kit,” on page 127.
 - Added the GetStoreDocInvalidFieldNum method, updated the StoreDoc method, and updated the STORE_DOC DDE command. (See Chapter 31, “Windows client customization,” on page 135.)

- Added an important note that you should read before attempting to schedule an offline database backup on Windows servers. See Chapter 32, “Windows server information,” on page 147.
10. 7.1.0.9 Tenth Edition September 13, 2002. Added or updated the following information:
- Added a note about specifying the **-S** parameter when running the ARSDOC program (see Chapter 9, “Command reference,” on page 27).
 - Added a note about DB2 UDB Version 7.2 (see Chapter 11, “DB2 UDB Version 7 information,” on page 39).
 - Added a note for customers using ODWEK with Java version 1.3.1 (see Chapter 13, “Hardware and software requirements,” on page 43).
 - Updated information about passwords (see Chapter 17, “Naming rules,” on page 57).
 - Added a section about character data conversion tables and the ICONVDEF program. Conversion tables are used to map characters in one code page to another. Added a note about the Traditional Chinese client incorrectly displaying the Description field for folders on a local server. See Chapter 18, “National language support,” on page 59).
 - Added a note about a change made in Version 7.1.0.7 to the OnDemand security function for userids. Updated information about passwords. See Chapter 20, “Password security,” on page 73.
 - Added an important note about configuring SMS table space file systems under Solaris (see Chapter 28, “Table space file systems,” on page 123).
 - Added a note for customers using ODWEK with Java version 1.3.1 (see Chapter 30, “Web Enablement Kit,” on page 127).
11. 7.1.0.10 Eleventh Edition October 31, 2002. Added or updated the following information:
- Updated ACIF information. New value for the EXTENSIONS parameter (see “EXTENSIONS parameter” on page 4), updated two messages (see “Messages” on page 7), and corrected one of the examples in the *Indexing Reference* (see “Using ACIF” on page 10).
 - Chapter 2, “Adobe software,” on page 11. Updated the information about Adobe viewing software.
 - Added a fix procedure for customers using the CICS® client to view AFP documents that are converted to line data. See “AFP to line data conversion” on page 13.
 - Added a note about viewing AFP documents on Windows XP. See “AFP viewing on Windows XP” on page 13.
 - Added a note about the **-h** and **-I** parameters. Added a note about date formats to the documentation on the ARSDATE program. Added a correction for the ARSMANT program about expiring data from cache storage. See Chapter 9, “Command reference,” on page 27.
 - Added a section on configuring the system so that the PDF indexer can locate Adobe font files (see “Specifying the location of Adobe fonts” on page 75).
 - New software support number (1-800-IBM-SERV). See Chapter 26, “Support,” on page 93.
 - Added a note about the requirement for using the TSM 32-bit API with OnDemand on 64-bit operating systems. (See Chapter 29, “TSM information,” on page 125.)

- Added a correction to the example of the `chgpassword` function for ODWEK (see Chapter 30, “Web Enablement Kit,” on page 127).
 - Added the `SetResourceCacheMode` method and updated the `RetrieveDoc` method. (See Chapter 31, “Windows client customization,” on page 135.)
 - Major revision to the information about the Xenos transforms, including support for HTML and XML output (see Part 3, “Xenos transforms,” on page 175).
12. Version 7.1.0.11 Twelfth Edition December 20, 2002. Added or updated the following information:
- Update to “EXTENSIONS parameter.” Added `EXTENSIONS=RESORDER` to “Running ACIF with inline resources.” Correction to “Understanding how ACIF processes fully composed AFP files.” See Chapter 1, “ACIF indexing,” on page 3.
 - Added a note about deleting input files. Added a note about the requirement for specifying the full path name of input files. See Chapter 12, “Generic indexer,” on page 41.
 - Added information that supports the new Java line data viewer. Set default value for `AUTODOCRETRIEVAL` parameter in the `ARSWWW.INI` file when using Internet Explorer. See Chapter 30, “Web Enablement Kit,” on page 127.
 - Added a special attribution for the Xenos d2e Platform. See Part 3, “Xenos transforms,” on page 175.
13. Version 7.1.0.12 Thirteenth Edition February 21, 2003. Added or updated the following information:
- Updated information about the `DEFAULT` keyword of the `FIELD` parameter. For the `PRMODE` parameter, added information about the new `SOSI4` value and added information about specifying an alphanumeric string. See Chapter 1, “ACIF indexing,” on page 3.
 - New information messages added to `ARSDOC` processing. Corrected the information about when the `-g` or `-G` parameter may be omitted when running the `ARSDOC` program. See Chapter 9, “Command reference,” on page 27.
 - Added information about using the CGI interface, the Java API, or the Java servlet with ODWEK. Added the `DOCSIZE` and `FOLDERDESC` parameters to the `ARSWWW.INI` file. Added the `_logoff=1` parameter to the `Add Annotation`, `Change Password`, `Document Hit List`, `Print Document`, `Retrieve Document`, `Update Document`, and `View Annotations` functions. See Chapter 30, “Web Enablement Kit,” on page 127.
14. Version 7.1.0.13 Fourteenth Edition April 30, 2003. Added or updated the following information:
- Updated the information about specifying the name of the instance to process with the `-h` and `-I` parameters. See Chapter 9, “Command reference,” on page 27.
 - Updated the information about backing up the OnDemand database by using the `ARSDB` program. See Chapter 9, “Command reference,” on page 27.
 - Correction to the Code Set for the CHS language code in HP-UX. Updated the DBCS Code Pages table. (See Chapter 18, “National language support,” on page 59.)
 - Added information about the enhanced tracing capability (`TRACEDD` parameter) now available for the PDF indexer (see “Trace facility” on page 77).

- Added a note about new versions of *IBM Content Manager OnDemand Version 7.1.: Messages and Codes*, SC27-1379-02 and *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Web Enablement Kit Implementation Guide*, SC27-1000-02. See Chapter 22, “Publications,” on page 83.
 - Updated edition of the ODWEK book is now available. The new edition contains all information previously included in these release notes from 7100 through 71012. See Chapter 30, “Web Enablement Kit,” on page 127.
 - Correction to the names of the Shared Library for the Java servlet. Added information about the new Auto Find function in the Java Line Data Viewer. The Auto Find function is activated or deactivated by using the ShowSearchString parameter in the ARSWWW.INI file. Added information about configuring ODWEK for DBCS languages. Added information about a new method (getNoteStatus) for the ODHit object and a new variation of the search method for the ODFolder object. See Chapter 30, “Web Enablement Kit,” on page 127.
 - Added information about Registry entries that are available to modify the behavior of the Windows client. See Chapter 31, “Windows client customization,” on page 135.
15. Version 7.1.0.14 Fifteenth Edition June 30, 2003. Added or updated the following information:
- Added information about using the ARSJESD program and a user-defined program to process additional job information sent by PSF through the APSUX15 user exit. See Chapter 6, “ARSJESD user exit,” on page 19.
 - A new parameter has been added the ARSLOG user exit interface. See Chapter 7, “ARSLOG user exit,” on page 23.
 - Added information about the new ARSSOCKD STOP command, used to stop the server. See Chapter 8, “ARSSOCKD stop command,” on page 25.
 - Added information about the removal of trailing blank characters from the names of applications, application groups, folders, printers, storage sets, distribution files, and distribution groups. See Chapter 9, “Command reference,” on page 27 and Chapter 17, “Naming rules,” on page 57.
 - Added information about support for DB2 UDB 8.1. See Chapter 10, “DB2 UDB Version 8.1 information,” on page 37.
 - Updated information about message number 90 (Bulk Data Retrieval). See Chapter 15, “Messages,” on page 49.
 - Added information about support for Oracle 9i. See Chapter 19, “Oracle information,” on page 65.
 - Added information about the new retrieval preview user exit. See Chapter 25, “Retrieval preview user exit,” on page 91.
 - Added a new section describing the OnDemand system tables. See Chapter 27, “System control tables,” on page 95.
 - Added ODWEK restriction: the AddFieldsToDocId parameter is not supported when accessing an OnDemand/390 Version 2 server. Added folder field name parameter to the Document Hit List API. Added five new methods to the ODCriteria object. Added a new variation to the logon method. Added information about how the AFP Web Viewer uses certain settings from logical views. Added information about an enhancement to the Image Web Viewer when viewing multi-page images. Added information to help solve a problem when using Internet Explorer 6.0 to view HTML pages that were generated by the AFP to HTML transform. See Chapter 30, “Web Enablement Kit,” on page 127.

16. Version 7.1.0.15 Sixteenth Edition September 9, 2003. Added or updated the following information:
 - Added support for Windows Server 2003. See Chapter 32, “Windows server information,” on page 147.

Part 1. General release notes

Chapter 1. ACIF indexing

Attributes of the input file

This section contains corrections to Chapter 5, User Exits and Attributes of the Input File, in the *Indexing Reference*, SC27-0842-00.

Pages 164 – 166 describe the data structure that is available from the ACIF user exits. The **cctype** (**CCTYPE**) parameter is now eight bytes in length.

- For UNIX and Windows, the **cctype** parameter should be defined as follows:
`char cctype[8]; /* CC type - A (ANSI), M (Machine), Z (ASCII) */`
- For OS/390, the **CCTYPE** parameter should be defined as follows:
`CCTYPE DS CL8 Carriage control type - A (ANSI) or M (Machine)`
- The **cctype** (**CCTYPE**) parameter begins at byte offset four and ends at byte offset eleven. The byte offsets of the **chars** (**CHARS**), **formdef** (**FORMDEF**), **pagedef** (**PAGEDEF**), **prmode** (**PRMODE**), and **trc** (**TRC**) parameters should be adjusted accordingly.

Concatenating resources to an AFP file

Note: Add the following section to Chapter 6, Hints and Tips, in the *Indexing Reference*, SC27-0842-00.

A resource group can be created and stored in a file by using the ARSACIF program. The resource file and the AFP file can then be concatenated together to form a file that can be processed by the indexing program. The following lists the parameters used to create a resource file using the ARSACIF program. The parameters process an AFP file named `credit.afp`, which is an AFP file that contains no indexing information or inline resources. The example is for an AIX system. For this example, the output file and the index file that ACIF usually generates are not needed; all resources are assumed to be in the directory named by the **USERLIB** parameter.

Contents of the ACIF parameter file `parms.acif`:

```
CC=YES
CCTYPE=A
RESTYPE=OVLY,PSEG,FDEF
INPUTDD=credit.afp
OUTPUTDD=/dev/null
INDEXDD=/dev/null
RESOBJDD=credit.res
USERLIB=/usr/resources
```

Command used to generate the resource group file:

```
arsacif parmdd=parms.acif
```

Command to concatenate the resource group file and the AFP file:

```
cat credit.res credit.afp > credit.out
```

You can then process the `credit.out` file with the indexing program, using the indexing information that was generated in the `credit.ind` file.

CONVERT parameter

ACIF can now generate page-level indexes **without** converting line data to AFP. Therefore, if you do not need to convert the input line data to AFP, but you do want ACIF to generate page-level indexes, then you should specify CONVERT=NO.

EXTENSIONS parameter

Beginning with Version 7.1.0.10, the EXTENSIONS parameter now supports the RESORDER value:

```
EXTENSIONS = RESORDER
```

When the RESORDER value is specified, inline resources do not have to appear in any particular order in the input file, although they must all appear before the beginning of the document. ACIF will read the inline resources into memory and use them when they are requested. **Note:** If there are many inline resources and little internal memory available, the system may run out of memory when using this option.

When the RESORDER value is not specified, inline resources must appear in the input file in the order in which they are used.

FIELD parameter

Note: When specifying the column number of the field, if the specified value exceeds the physical length of the record, ACIF reports an error condition and terminates processing, unless you specify a DEFAULT value.

Constant field

A *constant field* cannot be concatenated in an index with a field that is based on a floating trigger. A constant field is a field for which you specify the actual index value that will be stored in the database. It is possible to generate an index value by concatenating or combining the value that you specify for a constant field with the value that ACIF extracts from a document by using a trigger field. However, the trigger field cannot be based on a floating trigger.

DEFAULT keyword

Replace the second paragraph on page 71 of the *Indexing Reference*, SC27-0842-00 with the following:

The DEFAULT keyword supports the use of Download, which can be used to transmit data from the JES Spool to the OnDemand server. JES, by default, will truncate trailing blanks at the end of records. For example, assume that a report contains fixed length records, each 80 bytes in length. Columns 77 through 80 of the records contain audit data that is generated by a user-defined program. If a record has not been audited, the columns contain blanks. During file transmission, JES eliminates the blanks from the end of the records. When processing the records that have been truncated, ACIF will fail unless you specify a DEFAULT value. For more information regarding problems that can occur when using Download, see OnDemand support on the Web at <http://www.ibm.com/software/data/ondemand/mp/support.html> and search for document number 1005882.

MASK keyword

The MASK keyword of the FIELD parameter now supports a field that is based on a floating trigger. Previously, the MASK keyword could only be used with a transaction field.

Important: The field must be based on a floating trigger. An INDEX parameter that is based on the field cannot include any other fields and must not create grouprange or pagerange indexes.

Use the following syntax to specify a field with a mask when the field is based on a floating trigger:

```
FIELD n=record,column,length,(TRIGGER=n,BASE={0 |  
TRIGGER},MASK='@#=-^%')[,DEFAULT=value]
```

Where:

n

The field parameter identifier. When adding a field parameter, use the next available number, beginning with 1 (one).

record

The relative record number from the trigger on which the field is based. This is the record number where ACIF begins to search for the field. The supported range of values are from ±0 to 255.

column

The relative column number from the BASE (specified with the BASE keyword of the FIELD parameter). This is the column number where ACIF begins to search for the field. A value of 1 (one) refers to the first byte in the record. For files containing carriage-control characters, column one refers to the carriage-control. For those applications that use a specific carriage-control character to define page boundaries (for example, skip-to-channel one), consider defining the value of the carriage-control character as one of the TRIGGER parameters. If you specify BASE=0, then the *column* value can be from 1 to 32756. If you specify BASE=TRIGGER, then the *column* value can be from -32756 to 32756. If the specified value exceeds the physical length of the record, then ACIF reports an error condition and terminates processing.

length

The number of contiguous bytes (characters) that compose the field. The supported range of values are from 1 to 250. The field can extend outside the record length, if the column where it begins lies within the record length. In this case, ACIF adds padding blanks to fill out the record. If the field begins outside the maximum length of the record, then ACIF reports an error condition and terminates processing unless you specify a DEFAULT value.

TRIGGER=*n*

Identifies the trigger parameter that ACIF uses to locate the field. When using a MASK, you must specify a trigger that was defined with TYPE=FLOAT.

BASE={0 | TRIGGER}

Determines whether ACIF uses the starting column number of the trigger string value to locate the field data. Choose from 0 (zero) or TRIGGER. If you specify BASE=0, then ACIF adds zero to the field column offset. If you specify BASE=TRIGGER, then ACIF adds the starting column number of the trigger string value to the field column offset. You should specify BASE=0 if the field data always starts in a specific column. You should specify BASE=TRIGGER if the field data doesn't always start in a specific column, but is always offset

from the trigger string value a specific number of columns. For example, a trigger occurs in the second record on a page; the trigger string value can begin in any column in the record; a field based on this trigger occurs in the trigger record; the starting column number of the field is always ten bytes from the starting column number of the trigger; you would specify `BASE=TRIGGER` and a column offset of 10 (ten) so that ACIF correctly locates the field, regardless of the starting column of the trigger string value.

MASK= '@#=-~^%'

Specifies a pattern of symbols that ACIF uses to match data located in the field columns. If the data matches the MASK, then ACIF selects the field. You can specify the following symbols in the MASK:

- @ Matches alphabetic characters.
- # Matches numeric characters.
- = Matches any character.
- ~ Matches any non-blank character.
- ^ Matches any non-blank character.
- % Matches the blank character and numeric characters.

For example, given the following definitions:

```
TRIGGER2=*,25,'SOURCE',(TYPE=FLOAT)
FIELD2=0,38,4,(TRIGGER=2,BASE=0,MASK='####',DEFAULT=X'F1F0F0F0')
```

ACIF selects the field only if the data in the field columns contains numeric characters.

DEFAULT=*value*

Optionally use to specify a default index value when a record is not long enough to contain the field data. (However, if a record is not long enough and you do not specify a default value, ACIF will fail.) The default value can be specified as either character data or hexadecimal data. If the data to be specified is anything other than ASCII, then you must specify the default value as a hexadecimal value by using the format `X'constant'`, where constant is hexadecimal data (for example, `X'F1F0F0F0'`). The default value can be from 1 (one) to 250 characters in length. **Note:** Neither the administrative client nor ACIF validates the content of the data that you specify. Also, the MASK is not applied to the default value.

INDEX parameter

ACIF can now generate page-level indexes **without** converting line data to AFP.

INDEXOBJ parameter

The INDEXOBJ parameter now includes support for stapling on document boundaries when processing for Infoprint[®] Manager.

Important: This function should not be used with OnDemand.

ACIF normally removes any Begin/End Document structured fields from the input file and generates a single BDT/EDT for the entire output because MO:DCA indexes are relative to the Begin Document structured field. However, the stapling function uses BDT/EDT to indicate document boundaries for stapling. A new indexing option has been added to allow ACIF to pass through any BDT/EDT and

not create it's own. This file is suitable for printing, but should not be used with indexing because the resultant index will not be MO:DCA compliant and may not be processed correctly by programs which use the index, such as OnDemand.

To enable BDT/EDT pass through, specify the BDTLY option on the INDEXOBJ parameter. For example:

```
INDEXOBJ=BDTLY
```

MCF2REF parameter

Refer to page 88 of the *Indexing Reference*. In the first paragraph under **MCF2REF** and under **Options and values, CPCS**, delete the last sentence that begins with "If you map the AFP fonts ..." It is not true that you must specify MCF2REF=CPCS.

Messages

A new ACIF messages reference is available as part of *IBM Content Manager OnDemand: Messages and Codes, SC27-1379-01*. The new messages reference contains many new and updated messages and other important updates. All customers who use ACIF should now use the new messages reference instead of the messages reference that is contained in the *IBM Content Manager OnDemand for Multiplatforms: Indexing Reference*. You can obtain a copy of the *Messages and Codes* publication directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

The following messages have changed since SC27-1379-01 was released:

Message Number	New Message	Reason for Change	Example
APK440I	ACIF at <apar#> has completed normally with a Return Code 0.	The new message includes the current APAR number.	ACIF at PQ63028 has completed normally with a Return Code 0.
APK441I	ACIF at <apar#> has completed abnormally with a Return Code 4.	The new message includes the current APAR number.	ACIF at PQ65684 has completed abnormally with a Return Code 4.

Page-level indexes

ACIF can now generate page-level indexes **without** converting line data to AFP. Therefore, if you do not need to convert the input line data to AFP but you do want ACIF to generate page-level indexes, then you should specify CONVERT=NO.

PRMODE parameter

Beginning with Version 7.1.0.12, the PRMODE parameter now supports an additional value of SOSI4. Usage: PRMODE=SOSI4. SOSI4 is intended for use on workstation platforms where the user has DBCS text being converted from ASCII to EBCDIC, and is also using a PAGEDEF to convert the data to AFP. SOSI4 processing is similar to SOSI2, with the following difference. Specifying SOSI4 will cause ACIF to scan the input (EBCDIC) for SOSI characters and if any are found, they will be skipped but not counted as part of the input columns. This means that the PAGEDEF FIELD offsets should be correct after conversion from ASCII to EBCDIC and the user does not need to account for SOSI characters when

computing the PAGEDEF FIELD offsets. **Note:** The SOSI characters do have to be counted in determining the ACIF trigger and field offsets.

The PRMODE parameter also supports specifying an eight-byte alphanumeric string. The value is supplied to all of the ACIF user exits. Usage: PRMODE=aaaaaaaa, where aaaaaaaaa is the alphanumeric string.

RESTYPE parameter

The OBJCON keyword is valid with the RESTYPE parameter. Use the OBJCON keyword to specify that all object container files requested by the input data stream be included in the resource file. Usage: RESTYPE=OBJCON.

Running ACIF with inline resources

Note: Add the following section to Chapter 6, Hints and Tips, in the *Indexing Reference*, SC27-0842-00.

To successfully process an input file that contains inline resources, the inline resources must be included in the input file in the order in which they are used or EXTENSIONS=RESORDER must be specified. If a resource references another resource, the referenced resource must be included inline before the resource that references it. For example, if an overlay references a coded font that consists of the character set C0D0GT18 and code page T1D0BASE, the inline resources must be in this order:

```
code page T1D0BASE
character set C0D0GT18
coded font
overlay
```

ACIF does not look ahead in the inline resources, so, if the inline resources are not in the correct order, ACIF tries to read the referenced resource from a resource library. If the resource is not found, ACIF ends processing with an error.

Here is the recommended order that the resources should appear in the input file:

```
FORMDEF
CHARACTER SETS
CODE PAGES
PAGE SEGMENTS
OVERLAYS
PAGEDEF
```

TRIGGER parameter

Group triggers

In ACIF, a *group* is a named collection of sequential pages that form a logical subset of an input file. A group must contain at least one page; a group can contain all of the pages in an input file. However, most customers define their group triggers so that ACIF can logically divide an input file into smaller parts, such as by statement, policy, bill, or, for transaction data, number of pages. A group is determined when the value of an index changes (for example, account number) or when the maximum number of pages for a group is reached. ACIF generates indexes for each group in the input file. Because a group cannot be

smaller than one page, a group trigger should not appear more than once on a page. Please see the BREAK option of the INDEX parameter for more information about breaking groups.

In OnDemand, each indexed group of pages is known as a *document*. When you index an input file and load the data into the system, OnDemand stores the group indexes that are generated by ACIF into the database and stores the documents on storage volumes. OnDemand uses the group indexes to determine the documents that match the search criteria that is entered by the user.

Understanding how ACIF processes fully composed AFP files

Note: Add the following section to Chapter 7, ACIF Data Stream Information, in the *Indexing Reference*, SC27-0842-00.

Fully composed AFP files contain BNG and TLE Structured Fields in the following form:

```
BDT
  BNG
    TLE (group)
    ...
    ...
  BPG
    TLE (page - optional)
    ...
    ...
  EPG
ENG
...
...
EDT
```

When an input file contains BNG - ENG pairs or TLE Structured Fields, ACIF does not index the file. If you specify indexing parameters (such as TRIGGER, FIELD, or INDEX) for a file that contains TLE Structured Fields, then ACIF will fail with error message 462 - A trigger parameter was specified, but the input file is already indexed. If you specify indexing parameters for a file that contains BNG - ENG pairs, but does not contain TLE Structured Fields, ACIF will fail with error message 459 - Index needed for the groupname was not found.

ACIF processes a file containing BNG - ENG pairs and TLE Structured Fields in the following way:

1. For every BNG in the input, ACIF creates a group IEL Structured Field in the Index File.
2. ACIF makes a copy of the TLE Structured Fields from the input and places them into the Index File. The original TLE Structured Fields remain in the input file.

Therefore, the result of ACIF processing under these circumstances is the creation of an Index File. ACIF can complete normally but the load process into OnDemand may still fail if the format of the input file is incorrect:

- If the input file contained BNG - ENG pairs with no group level TLE Structured Fields between them, then the load process will fail with the message: 0 fields submitted, n expected, where n is the number of fields defined to OnDemand.
- If the input file does not contain any BNG - ENG pairs, then the load process may run out of memory looking for the start and end of the groups.

Using ACIF

Figure 18 on page 58 of the *Indexing Reference*, SC27-0842-00 has an error. The parameter for the unique group names is incorrectly specified as UNIQUEBNQS. The correct parameter for the unique group names is UNIQUEBNGS. (The character G was incorrectly specified as the character Q in the figure.)

Writing inline resources to the output file

Note: Add the following section to Chapter 6, Hints and Tips, in the *Indexing Reference*, SC27-0842-00.

When you are indexing and writing inline resources to the output document file, the offsets in the index object file are the same as if you are doing regular resource collection to a resource file. This is because the offsets are calculated from the Begin Document (BDT) structured field, not from the beginning of the output document file. The offset from the BDT structured field to the indexed data is the same regardless of whether resources precede it.

Chapter 2. Adobe software

This section contains the latest information about the Adobe software that can be used with OnDemand.

Notes:

1. IBM recommends that you use the full Adobe Acrobat product or Adobe Acrobat Approval to view PDF documents from the Windows client or work with PDF input files from the administrative client. Adobe software is not provided by IBM; you should purchase Adobe Acrobat or Adobe Acrobat Approval from Adobe or some other software vendor.
2. If you are using the OnDemand Web Enablement Kit (ODWEK) to view PDF documents from Internet Explorer 5.5 with SP2 or later or Internet Explorer 6, you should update your Acrobat viewing software to Adobe Acrobat Version 5 or later. The Acrobat Version 5 software contains ActiveX controls that are required by Internet Explorer 5.5 with SP2 or later and Internet Explorer 6. See <http://www.adobe.com> for information about Acrobat viewing software.

The following Adobe software can be used with OnDemand:

- Adobe Acrobat. Licensed software to view and create PDF files. You can use Adobe Acrobat to do the following:
 - Integrate Adobe Acrobat with the Windows client. If the Data Type of the OnDemand application is PDF, when a user retrieves a PDF document from the system, the client loads the PDF document into the client viewing window. IBM recommends that you purchase Adobe Acrobat or Adobe Acrobat Approval for your users that need to view PDF documents from the Windows client.
 - Integrate Adobe Acrobat with the administrative 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 PC from which you plan to run the administrative client. IBM recommends that you purchase Adobe Acrobat or Adobe Acrobat Approval for your users that need to work with PDF input files from the administrative client.

Important: You should install Adobe Acrobat **before** you install the OnDemand client. OnDemand provides the ARSPDF32.API file to enable PDF viewing from the clients. If you install the clients after you install Adobe Acrobat, then the installation program will copy the API file to the Acrobat plug-in directory. If you install the clients before you install Adobe Acrobat, then you must copy the API file to the Acrobat plug-in directory. 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, where x.y is the version of Acrobat, for example, 4.0, 5.0, and so forth.

- Adobe Type Manager (ATM). Licensed software to manage Adobe Type 1 fonts. If your users need to view AFP documents that use the IBM Core Outline Fonts or the Sonoran Metric Outline Fonts that are provided with OnDemand, then they will need ATM installed on their systems. On Windows 2000 systems, ATM is part of the base operating system. If you plan to use the OnDemand Windows

client with other operating systems, then you should obtain the appropriate version of the ATM software from Adobe.

- Adobe Reader. Free of charge software (available for download from the Adobe Web site) to view PDF documents. The OnDemand client can start Adobe Reader if the Data Type of the OnDemand application is User Defined, the File Type of the OnDemand application is PDF, and Adobe Reader is associated with the PDF file type on the client operating system.

You can find out more about Adobe software from Adobe on the Web at:

<http://www.adobe.com>

Chapter 3. AFP information

AFP to line data conversion

Version 7.1.0.10 contains a fix to the AFP to line data conversion routine. Prior to Version 7.1.0.10, AFP data being converted to line data for viewing by the CICS client was incorrectly converted from code page 1208 to code page 500. (The correct conversion should have been from code page 850 to code page 500.) Use the information in Table 1 to determine if your system is affected and to resolve the problem.

Table 1. AFP to Line Data Conversion - problem, determination, action

Problem	The code page conversion of AFP to line data is incorrect when viewing data using the CICS client. AFP data being viewed by the CICS client is being incorrectly converted from code page 1208 to code page 500. This will cause text to be displayed incorrectly by the CICS client. The correct conversion is from code page 850 to code page 500.
Determination	You are affected by this problem if you are trying to view AFP data from a CICS client and the application has been added with a version of OnDemand after 2.2.1.9 and before 7.1.0.10.
Action	<p>After installing the 7.1.0.10 PTF, the application that is defined for the AFP data must be deleted and then added back to the server. A copy of the application can be saved and used to create the new application so that the parameters do not have to be entered again. The following steps are required to correct the problem by using the OnDemand administrative client.</p> <ol style="list-style-type: none">1. Start the OnDemand administrative client and logon to the server where the AFP data is loaded.2. Make a temporary copy of the application group that contains the AFP data.3. Make a temporary copy of the application that contains the AFP data, using the original application. Assign the temporary application to the temporary application group.4. Delete the original application.5. Make a copy of the temporary application and assign it to the original application group.6. Delete the temporary application group.7. Logon to the CICS client to view the data. <p>See the online help in the OnDemand administrative client for more information on these steps. If you have questions about this procedure or if you encounter any problems, please contact the IBM support center.</p>

AFP viewing on Windows XP

The AFP Viewer (which is used in the Windows client and the AFP Web Viewer) does not support the Clear Type Font support that is available in Windows XP.

Chapter 4. AIX information

AIX 4.3.3 or later

For OnDemand servers that are running AIX 4.3.3 or later, you must install the C Set ++ Runtime for AIX 4.3 (x1C.rte 5.0.0.0) before you install or upgrade to OnDemand Version 7.1. You can get the latest version of this software from IBM Software on the Web at:

`ftp.software.ibm.com/aix/products/ccpp/vacpp5_runtime.tar.Z`

AIX Fix Pack 9

If you upgrade AIX to Fix Pack 9, depending on your use of OnDemand, you may have intermittent problems with ARSSOCKD and ARSLOAD, causing you to restart the system or causing loads to fail.

After upgrading to Fix Pack 9, if you experience problems with ARSSOCKD and ARSLOAD, do the following:

1. Bring the following file sets to current levels:

- `bos.mp`
- `bos.rte.libpthread`
- `bos.rte.libc`

You can download the file sets from IBM Fix Distribution on the Web at <http://service.software.ibm.com/rs6k/fixdb.html>.

Note: After you update the file sets, you will need to reboot the machine, because the libraries will still be present in shared memory until they are flushed out by a reboot. However, before rebooting, continue with the following steps.

2. Add the following environment variables to the `/etc/environment` file:

- `AIXTHREAD_MNRATIO=1:1`
- `AIXTHREAD_SCOPE=S`

Note: You can update the current running environment by using the `export` command from the AIX command line to set the environment variables. For example:

```
export AIXTHREAD_MNRATIO=1:1
export AIXTHREAD_SCOPE=S
```

However, in addition to using the `export` option, you should also update the `/etc/environment` file so that the environment variables will automatically be set by the system the next time that the system is restarted.

3. Save the file.
4. Reboot the server.

Chapter 5. ARS.CFG

For UNIX servers, the following parameters have been changed in or added to the ARS.CFG file in Version 7.1.

ARS_DB_ENGINE

Use to specify the database manager. See **DB_ENGINE**.

ARS_MESSAGE_OF_THE_DAY

Use to show the message of the day. Set to the full path name of a file that contains the message that you want to show. For example:

```
ARS_MESSAGE_OF_THE_DAY=/opt/ondemand/tmp/message.txt
```

The contents of the message file can contain a maximum of 1024 characters of text. The administrative client and the Windows client show the message after the user logs on to the server. To close the message box and continue, the user must click OK. If you do not specify a message file, then the normal client processing occurs.

ARS_NUM_DBSRV

The default value is now 4 (four).

ARS_ORACLE_HOME

Use to specify the base installation directory for Oracle. The default value is:

```
ARS_ORACLE_HOME=/oracle
```

Replace the string `/oracle` with the name of the directory in which Oracle was installed.

ARS_STORAGE_MANAGER

A new option **TSM** has been added. You should specify `ARS_STORAGE_MANAGER=TSM` to link the server program to an archive storage manager. The **ADSM** option will continue to be supported for existing customers.

DB_ENGINE

A new parameter **ARS_DB_ENGINE** has been added to specify the database manager. The **DB_ENGINE** parameter will continue to be supported for existing customers.

Chapter 6. ARSJESD user exit

This section contains additional information for Chapter 20. Download exits and processing, in the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840-00.

ARSJESD processing

The ARSJESD program is the component of Download for OS/390 that runs on the workstation. The `-x` parameter of the ARSJESD program may be used to specify the name of a user-written program to process additional job information sent by PSF through the APSUX15 user exit.

If the ARSJESD program was invoked with the `-x` parameter, it calls the specified user-written program. The ARSJESD program passes the file name and the additional job information to the user-written program.

The additional job information is installation dependent. See *IBM Print Services Facility™ for OS/390 and z/OS: Download for OS/390*, S544-5624 for details about the APSUX15 user exit and the content, format, and purpose of the additional job information. The processing done by the user-written program is also installation dependent. See your Infoprint Manager or PSF information for information about processing the additional job information with a user-written program.

Sample user exit program

In OnDemand, it is possible to use the ARSJESD exit point in order to customize and enhance the standard functionality within the product. A user exit is a point during processing that enables you to run a user-written program and return control of processing after your user-written program ends.

The ARSJESD exit point is a server exit. OnDemand provides data at the ARSJESD exit that serves as input to the user-written program. The data consists of the file name and the additional job information. Using this exit, it is possible to do functions such as parse the additional job information that is sent by PSF and rename the input file by using one of the PSF parameters.

The program invoked at the ARSJESD exit point is defined by specifying the `-x userProgram` parameter when starting the ARSJESD program, where *userProgram* is the name or full path name of the user-written program.

The following example demonstrates a common use for the ARSJESD exit point. The example processes the file name and the additional job information. The example renames the input file by using the value of the WRITER parameter that was sent by PSF. For the sake of simplicity, the sample is not demonstrated across all of the supported platforms. IBM recognizes that the scripting languages between platforms do vary, but the principles which are described here are uniform across all supported platforms; only the syntax differs. The sample is provided for an AIX system.

Note: The example program 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.

```

#!/bin/ksh
#
# COPYRIGHT: 5697-G34 (C) COPYRIGHT IBM CORPORATION 2003
#           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.
#
# MODULE:   writer_rename
#
# DESCRIPTION:
# This Korn shell script is designed as a exit routine to be executed by
# the ARSJESD program after a file has been downloaded from OS/390.
# This script parses and prints the additional job options that were sent
# by PSF and renames the file name by using the value of the WRITER parameter.
#
#
# set -x
#
ME=$(basename $0)
PRODUCT="OnDemand"
PRODUCT_DIR=/usr/lpp/ars
ARS_BIN_DIR=${PRODUCT_DIR}/bin
AWK=/bin/awk
DATE=/bin/date

status_msg ()
{
    print "\n${DATE} -- ${ME} -- \nERROR: Unable to determine WRITER field"
    print "\t\t$1"

    return
}

parse_objects ()
{
    while getopts o: i
    do
        print ${OPTARG} | ${AWK} -F [=] '{print $1}' | read option
        case $option in
pa) value=${OPTARG#*=}
            while [ "$value" != "$oldvalue" ]
            do
                oldvalue=$value
                arg=${value%*=*}
                value=${value#*=}
                case $arg in
                    segmentid ) SEGID=${value%*,*} ;;
                    forms      ) FORM=${value%*,*} ;;
                    class      ) CLASS=${value%*,*} ;;
                    destination ) DEST=${value%*,*} ;;
                    writer      ) WRITER=${value%*,*} ;;
                esac
                value=${value#*,}
            done ;;
*) print ${OPTARG} | ${AWK} -F [=] '{print $1, $2}' | read arg value
            case $arg in
                fileformat ) FORMAT="$value" ;;
                datat      ) DATAT="$value" ;;
                chars      ) CHARS="$value" ;;
                cc          ) CC="$value" ;;
                cctype     ) CCTYPE="$value" ;;
                pagedef    ) PAGEDEF="$value" ;;
                prmode     ) PRMODE="$value" ;;
                trc        ) TRC="$value" ;;
                cop        ) COPIES="$value" ;;
                datac      ) DATAC="$value" ;;
                f          ) FORMDEF="$value" ;;
                outbin     ) OUTBIN="$value" ;;
                jobn       ) JOBNAME="$value" ;;
            esac
        esac
    done
}

```

```

        us        ) USER="$value" ;;
        no        ) MVSNAME="$value" ;;
        pr        ) TEXT="$value" ;;
        address1  ) ADDRESS1="$value" ;;
        address2  ) ADDRESS2="$value" ;;
        address3  ) ADDRESS3="$value" ;;
        address4  ) ADDRESS4="$value" ;;
        bu        ) BLDG="$value" ;;
        de        ) DEPT="$value" ;;
        na        ) NAME="$value" ;;
        ro        ) RO="$value" ;;
        ti        ) TITLE="$value" ;;
    esac ;;

done
}

FILENAME="$1"
shift
print "\n${DATE} -- ${ME} -- \nStart writer_move for file ${FILENAME}\n"
print "File Name = ${FILENAME}"

print ${FILENAME} | awk -F [.] '{print $1, $2, $3, $4, $5, $6, $7}' | read MVS JOBNAME DATASET FORMS YYDDD HHMMM EXT

PARMS=$(print $@ | tr -d '\ ' )
print "Parameters = ${PARMS}"

parse_objects ${PARMS}

APPGRP=${WRITER}

if [ "${WRITER}" != "" ]
then
    status_msg "WRITER field specified."
    NEWFILENAME="${MVS}.${JOBNAME}.${WRITER}.${FORMS}.${YYDDD}.${HHMMM}.${EXT}"
else
    status_msg "WRITER field not specified."
    NEWFILENAME="${MVS}.${JOBNAME}.${DATASET}.${FORMS}.${YYDDD}.${HHMMM}.NOWTR"
fi

print "File renamed to ${NEWFILENAME}"

mv ${FILENAME} ${NEWFILENAME}

```

Chapter 7. ARSLOG user exit

Beginning with Version 7.1.0.14, a new parameter has been added to the ARSLOG user exit interface. The new parameter specifies the name of a temporary file that contains the document that was stored in the system log. The system deletes the temporary file after the call to the ARSLOG user exit is complete.

User action required: Verify any user-defined ARSLOG script files, batch files, and programs.

Sample ARSLOG user exit batch file for Windows:

```
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 %9 - Document File
REM

ECHO %1 %2 %3 %4 %5 %6 %7 %8 %9>%ARS_TMP%\System.log

REM make sure the %ARS_TMP%\Syslog directory exists
IF EXIST %9 COPY %9 %ARS_TMP%\Syslog

EXIT
```

Sample ARSLOG user exit script for UNIX:

```
# $1 - OnDemand Instance Name
# $2 - Time Stamp
# $3 - Log Identifier
# $4 - Userid
# $5 - Account
# $6 - Severity
# $7 - Message Number
# $8 - Message Text
# $9 - Document File
#

echo "$@" >> ${ARS_TMP}/syslog.log

if [ -n "$9" ];then
  if [ -f "$9" ];then
    print "Copy log doc $9\n" >> ${ARS_TMP}/syslog.log
    cp $9 /tmp/syslogdocs/${basename $9}.doc 2>> ${ARS_TMP}/syslog.log
  else
    print "$9 does not exist\n" >> ${ARS_TMP}/syslog.log
  fi
fi

exit 0
```

Chapter 8. ARSSOCKD stop command

Beginning with Version 7.1.0.14, IBM now provides a command to stop the server process on UNIX servers. The syntax is:

```
arssockd stop instance
```

Where *instance* is the name of the OnDemand instance to stop. Instance names are listed in the ARS.INI file. The default instance name is archive.

Notes:

1. When you stop the server process, all users who are connected over the network to the OnDemand system are disconnected. Therefore, it is a good idea to warn connected users before stopping the server process.
2. The ARSSOCKD STOP command is for UNIX servers only. For Windows servers, the server process runs as a service. The service can be stopped by using the Services administrative tool in Windows or the OnDemand configurator client.

Chapter 9. Command reference

This section contains corrections and updates to the Command Reference part of the *Administrator's Guide*, SC27-0840-00.

General information

Removal of trailing blank characters: Beginning with Version 7.1.0.14, trailing blank characters are trimmed from the name when adding or updating an application, application group, folder, printer, storage set, distribution file, or distribution group. As a reminder, for existing objects, the name specified to a command line program must include trailing blank characters that are part of the name.

Use of the `-h` and `-I` parameters: The OnDemand programs use the `-h` and `-I` parameters to determine the name of the OnDemand instance to process. You must specify the parameter and name the **instance** if any of the following are true:

- The name of the default instance is ARCHIVE and you are using a port number other than the default port number (1445).
- The name of the default instance is not ARCHIVE.
- You are running more than one instance on the same workstation and you want to process an instance other than the default instance.
- You are running the program from an object server that resides on a different workstation than the library server.

The programs locate the specified instance name in the ARS.INI file (for UNIX servers; the Registry for Windows servers) 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.

Use of quoting in commands: When you run an OnDemand program from the command line and you specify parameter values that contain a null (blank) character or some other special character (such as the parenthesis), if the command line interpreter of the operating system from which you are running the command requires the quoting of special characters, then you must delimit the parameter value with double quote characters. For example, when running the ARSDOC GET program from the command line under UNIX and you specify the `-o` parameter to specify one or more database field names, then in addition to delimiting each field name with parenthesis, you must also delimit the entire parameter value with double quote characters: `-o "(sdate)(student)".` However, if you were to run the same program under Windows, then you would not need the double quote characters: `-o (sdate)(student).` Please see the operating system documentation for more information about quoting and running commands. **Note:** There are some exceptions to this rule, where quotes may always be required. Please see the *Administrator's Guide* and these release notes to identify specific instances where parameters and values may always require quoting.

ARS_ADSM

The ARS_ADSM shell script can be used to invoke the TSM processes that start and stop the TSM server and backup the TSM database. However, the ARS_ADSM shell script has not been thoroughly tested under all conditions. For example, the ARS_ADSM shell script does not monitor the TSM processes or report on the success or failure of the TSM processes. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of the program. **Important:** If you use the ARS_ADSM shell script to start the TSM process that backs up the TSM database, then you should query the TSM server to determine whether the backup was successful before you continue with other OnDemand or TSM work. See your TSM documentation for details.

ARSADM

Deleting applications, application groups, and folders

Beginning with version 7106, the ARSADM program now supports the deletion of applications, application groups, and folders. The ARSADM program will prompt the user before each delete occurs, unless the `-x` parameter is specified on the command line. If the `-x` flag is specified, then the ARSADM program will automatically delete the specified objects.

The delete command for the USER, GROUP, PRINTER, and SET functions now supports the `-x` parameter to automatically delete the specified objects; otherwise, the user will be prompted before each delete request occurs.

Examples:

- Delete a list of application groups:

```
arsadm applgrp -u userid -p passwd -h librSrvr -F applgrpdel.lis -x -v
```

The contents of `applgrpdel.lis` are:

```
# Delete applgrps #
[-n "dkw-applgrp1"] [-a d]
[-n "dkw-applgrp2"] [-a d]
[-n "dkw-applgrp3"] [-a d]
[-n "dkw-applgrp4"] [-a d]
```

All four application groups (and all of the applications that are contained in the application groups) will be deleted without prompting the user.

- Delete a list of application groups:

```
arsadm applgrp -u userid -p passwd -h librSrvr -F applgrpdel.lis -v
```

The contents of `applgrpdel.lis` are:

```
# Delete applgrps #
[-n "dkw-applgrp1"] [-a d]
[-n "dkw-applgrp2"] [-a d]
```

A prompt will occur before each application group is deleted. In addition, a message will be displayed that indicates the number of documents that are in the application group. For example:

```
01/15/02 16:15:19: '0' documents are loaded in the application group.
```

```
Deleting 'dkw-applgrp1'... Are you sure (Y/N)? y
```

01/15/02 16:15:19: '1550' documents are loaded in the application group.

Deleting 'dkw-applgrp2'... Are you sure (Y/N)? y

- Delete an application from an application group:

```
arsadm appl -u userid -p passwd -h librSrvr -F appldel.lis -x -v
```

The contents of appldel.lis are:

```
# Delete appls #  
[-n "dkw-app11"][-a d][-g "dkw-applgrp1"]  
[-n "dkw-app12"][-a d][-g "dkw-applgrp2"]
```

Two applications will be automatically deleted (one from each application group); the user will not be prompted.

- Delete a list of folders:

```
arsadm folder -u userid -p passwd -h librSrvr -F folderdel.lis -x -v
```

The contents of folderdel.lis are:

```
# Delete folders #  
[-n "dkw-folder1"][-a d]  
[-n "dkw-folder2"][-a d]
```

Two folders will be deleted; the user will be prompted. For example:

Deleting 'dkw-folder1'... Are you sure (Y/N)? y

Deleting 'dkw-folder2'... Are you sure (Y/N)? y

SET function

The SET function now supports storage nodes on an OnDemand for z/OS and OS/390 V7.1 object server.

The **-T** parameter designates the node type. The choices are:

- T **n** The storage node is not on a z/OS object server. This is the default value, except on an z/OS server.
- T **o** The storage node is on a z/OS object server and the Access Method is OAM. This is the default value on a z/OS server.
- T **v** The storage node is on a z/OS object server and the Access Method is VSAM.

The **-N** parameter specifies the Storage Node Name, Collection Name, or High Level Qualifier. The choices are:

-N storagenodeName

The storage node is not on a z/OS object server.

-N collectionName

The storage node is on a z/OS object server and the Access Method is OAM.

-N highLevelQualifier

The storage node is on a z/OS object server and the Access Method is VSAM.

The following example shows how to add the storage set Set1 with a storage node identifier of node1. The object server name is objserver and the storage node is non-cache with an Access Method of OAM (the default value). The Collection Name in the administrative client is node1.

```
arsadm set -u user -p passwd -h od390server -n Set1 -N node1 -H objserver -v
```

The following example shows how to update the storage set Set1 with a storage node identifier of node1. The object server name is objserver and the storage node is non-cache with an Access Method of OAM (the default value). The Collection Name in the administrative client is node1.

```
arsadm set -u user -p passwd -h od390server -n Set1 -N node1 -T v -A u -a u -v
```

Notes:

1. For a storage node that is located on a z/OS object server, the Logon and Password values cannot be specified regardless of whether the node is cache only or non-cache.
2. When logging on to an OS/400® server, the Logon and Password values cannot be specified for a storage node regardless of whether the node is cache only or non-cache.

USER function

The USER function now supports the **-E** parameter. You can use the **-E** parameter to specify the user's e-mail address. For example:

```
arsadm user -E ondemand@us.ibm.com
```

The e-mail address that you specify can be from one to 254 characters.

ARSADMIN

The **-h** parameter is now required for all functions except compress and decompress.

The ARSADMIN program now supports the ICONV function. The syntax of the ICONV function is:

```
arsadmin iconv -s sourceFile -F sourceCodeSet -o outputFile -T outputCodeSet
```

Where:

- s sourceFile specifies the name of the file that contains the document that is in the code set specified by the **-F** parameter
- F sourceCodeSet specifies the code set for the document specified by the **-s** parameter
- o outputFile specifies the name of the file that will contain the document after it has been converted to the code set specified by the **-T** parameter
- T outputCodeSet specifies the code set for the document specified by the **-o** parameter

The following example shows how to use the ICONV function to convert the input.utf8 file from Unicode to EBCDIC and save the output in the output.ibm500 file.

```
arsadmin iconv -s input.utf8 -F UTF-8 -o output.ibm500 -T IBM-500
```

ARSDATE

Table 8 on page 266 of the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840-00 lists the standard date and time formats that are supported by OnDemand. If the input data contains a date or time format that is not listed in the table, you can specify the format to OnDemand. However, when specifying a format, you can only use values and separators from

the standard formats that are listed in the table. Date and time formats may also be specified on the Load Information page in applications and the Field Information page in folders.

ARSDB

The ARSDB program can be used to backup the OnDemand database to various devices and optionally, TSM storage. The options to backup the database are specified by using the **-y**, **-Y**, **-z**, and **-Z** parameters. This section describes changes to these parameters and updates to the ARSDB reference documentation in *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840.

Note: The note that appears on page 272 and 273 of *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840 regarding the use of a manual tape device is no longer valid and may be ignored.

Specifying backup devices

Note: The information in this section pertains to backing up the database to devices that are managed by the operating system. See “Backing up to TSM storage” for information about backing up the database to TSM storage.

Beginning with Version 7.1.0.13, the maximum number of backup devices that may be specified with the ARSDB program is now 100. To specify backup devices, use the following format:

```
arsdb [-y | Y | -z | Z] device1 [,device2, ... ,device100]
```

For example, the following shows how to initiate an online backup of the database by using three devices:

```
arsdb -z device1,device2,device3
```

Where:

-z is the backup option. See the ARSDB program reference in *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840 for a description of the backup options.

device1,device2,device3 is a comma-separated list of devices to which the database manager will concurrently write the backup image of the database (or tablespace, depending on the backup option). Examples:

```
/dev/rmt0,/dev/rmt1,/dev/rmt2
```

```
\\.\Tape0,\\.\Tape1,\\.\Tape2
```

```
/arsdb/backup1,/arsdb/backup2,/arsdb/backup3
```

Backing up to TSM storage

DB2 provides support for backing up the OnDemand database to TSM storage. The ARSDB program may be used to backup the database by using multiple connections to TSM. Multiple connections to TSM can improve the performance of the backup task and reduce the time that it takes to complete the backup. **Note:** Before you can backup the database to TSM storage, you must configure OnDemand and TSM. See the *OnDemand Installation and Configuration Guide* for your server operating system. The *DB2 Administration Guide* contains more information about TSM support.

Use the following format of the ARSDB program:

```
arsdb -[y | Y | z | Z] ADSM [,n]
```

For example, the following shows how to initiate an online backup of the database to TSM storage by using three connections to TSM:

```
arsdb -z ADSM,3
```

Where:

-z is the backup option. See the ARSDB program reference in *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840 for a description of the backup options.

ADSM specifies that the backup image file will be created in TSM storage.

3 is the number of connections that DB2 establishes with TSM. The number can be from 1 (one) to 10 (ten).

ARSDOC

ADD function

When using the ADD function and specifying field values with the **-n** parameter:

- Specify a null (blank) field value by using single quotes within double quotes. For example: `-n middle=""`
- Specify a string field value that contains a null (blank) or other special character by enclosing the field value in single quotes within double quotes. For example: `-n name=""Sally Smith""`

-f parameter

The **Note** on page 286 of the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840-00 applies only when an application group name is not provided.

GET function

The GET function can now sort the data that is retrieved from the server. The data is sorted by using the Sort field from the Field Information page for the folder.

Note: For the GET function, you must specify the **-n** parameter, so that the documents will be returned in the sorted order.

Messages

Beginning with Version 7.1.0.12, the ARSDOC program can print the PTF version number and the ARSDOC GET function can print the number of documents that were queried and retrieved and print a status message for each document that is retrieved.

To enable the messages, you must specify the **-v** parameter.

The following shows an example of the new messages:

```
12/08/02 10:33:36: Starting arsdoc. Version: 7.1.0.12
12/08/02 10:33:36: arsdoc get -u admin -h instance -q 000-000-000 -f CC Stmtms -o test -v -c
12/08/02 10:33:36: Attempting login for userid 'admin' on server 'instance' ...
12/08/02 10:33:37: Login successful
```

```
12/08/02 10:33:37: Searching for folder 'CC Stmts' ...
12/08/02 10:33:38: Search successful
12/08/02 10:33:38: Searching for documents in 'CC Stmts' ...
12/08/02 10:33:40: Search successful
12/08/02 10:33:40: 2 document(s) have been queried. Retrieving 2 document(s).
12/08/02 10:33:41: (1): Retrieving document for userid 'admin' ...
12/08/02 10:33:41: Document successfully retrieved and stored in file 'test'
12/08/02 10:33:42: (2): Retrieving document for userid 'admin' ...
12/08/02 10:33:42: Document successfully retrieved and stored in file 'test'
12/08/02 10:33:45: arsdoc completed.
```

QUERY function

The QUERY function can now sort the data that is retrieved from the server. The data is sorted by using the Sort field from the Field Information page for the folder.

-S parameter

A date format string can now be specified with the -S parameter. For example:

```
-S "01011990,12311990,%m%d%Y"
```

Where %m%d%Y is the date format. See the ARSDATE program reference in the Administrator's Guide for a list of the standard date formats. If you do not specify a date format, then the date values that you specify must be in the date format that was selected for the Display Format field on the Field Information page for the folder.

Notes:

1. The -S parameter enables you to specify a date range that the ARSDOC program uses to limit a search to specific tables. When you specify this parameter, the ARSDOC program searches only tables that contain a segment within the specified date range. For most queries, you should **always** specify this parameter and a date range. Doing so limits the range of a query and can significantly improve the performance of a query.
2. For the ADD function, if you specify the -o parameter, you cannot specify the -S parameter.
3. For all functions, if you specify the -q parameter, you cannot specify the -S parameter.

UPDATE function

Beginning with version 7.1.0.7, if the folder that is specified with the -f parameter contains only one application group, then you can omit the -g or -G parameter (you no longer have to specify the name of the application group).

When using the UPDATE function and specifying field values with the -n parameter:

- Specify a null (blank) field value by using single quotes within double quotes. For example: `-n middle=""`
- Specify a string field value that contains a null (blank) or other special character by enclosing the field value in single quotes within double quotes. For example: `-n name=""Sally Smith""`

-x parameter

The ARSDOC program now supports the `-xloadId` parameter.

- For the GET function, use the `-x` parameter to limit the documents that can be retrieved to the set of documents that were loaded into the system under the specified `loadId`.
- For the QUERY function, use the `-x` parameter to limit the query to the set of documents that were loaded into the system under the specified `loadId`.

When you specify the `-x` parameter, use the `-f` and `-G` parameters as follows:

- Specify the `-f` parameter to search all application groups. You can specify the search using the `-i` parameter or the `-q` parameter.
- Specify the `-G` parameter to search a specific application group. You must specify the search using the `-i` parameter.
- Specify both the `-f` parameter and the `-G` parameter. The ARSDOC program will verify that the application group can be searched from the folder.

When you specify the `-x` parameter, you cannot specify the `-X` parameter.

The following example shows how to limit a query to the documents that were loaded into the system under a specific load ID. For example, assume that the specified folder could be used to search several application groups; each application group contains more than one application; there are 500,000 documents in the application groups. By using the `-x` parameter, the query will be limited to the set of documents that was loaded into the system under the specified load ID. Without the `-x` parameter, the query is run against all 500,000 documents.

```
arsdoc query -h rockies
-x 19867-025-0-3FAA-10136-10136
-f load-scanned-images -q query -o output -v
```

-X parameter

The ARSDOC program now supports the `-XloadId` parameter.

- For the GET function, use the `-X` parameter to retrieve documents by using the index file that was generated for the specified `loadId`.
- For the QUERY function, use the `-X` parameter to build a hit list from the index file that was generated for the specified `loadId`.

When you specify the `-X` parameter, you must specify the `-G` parameter and name the application group. When you specify the `-X` parameter, you cannot specify the `-x` parameter, or the `-i`, `-q`, `-S`, and `-f` parameters.

The following example shows how to use an index file to retrieve documents from the system. The index file was generated for the set of documents that was loaded into the system under the specified load ID. When using the `-X` parameter, the database is not queried; rather, documents are retrieved based on the information in the index file.

```
arsdoc get -h rockies
-X 19867-025-0-3FAA-10136-10136
-G load-scanned-images-jpeg -o jpeg -v
```

ARSLOAD

The syntax diagram on page 305 shows the **-X** parameter as a required parameter. The **-X** parameter is optional and is used only to specify an Indexer that is different than the Indexer that is specified on the Indexing Information page.

ARSMaint

The information on page 319 of the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Administrator's Guide*, SC27-0840-00 incorrectly states that the ARSMaint program deletes (expires) files from a cache storage file system starting with the files with the oldest date. However, the ARSMaint program does not sort files to be expired by date. Instead, the ARSMaint program may expire any data beyond the date that is specified, up to the threshold specified.

ARSTBLSP

The first paragraph under Notes on page 325 in the *Administrator's Guide*, SC27-0840-00 that begins with "Before you run the ARSTBLSP program..." no longer applies and may be ignored.

Chapter 10. DB2 UDB Version 8.1 information

Beginning with Version 7.1.0.14, OnDemand now supports DB2 UDB Version 8.1 Enterprise Server Edition. See <http://www-3.ibm.com/software/data/db2/udb/sysreqs.html> for a list of the system requirements for your operating system.

OnDemand has been tested with DB2 UDB 8.1 on the following operating systems:

Operating System	Version
AIX	AIX 4.3.3 with maintenance level 9
HP-UX	HP-UX 11i with December 2001 Goldbase and Goldapps 11i
Solaris	Solaris 8 with recommended and security patch set
Windows	Windows 2000 Server Windows Server 2003 Enterprise Edition

The new DB2 Version 8 Information Center contains everything you need to know about DB2 Universal Database installation, configuration, administration, application development, and more. See http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8pubs.d2w/en_main for browser requirements for and links to the information center.

See *IBM DB2 Universal Database Version 8: Quick Beginnings for DB2 Servers*, GC09-4836 for information about system requirements, installation instructions, and other information.

Chapter 11. DB2 UDB Version 7 information

Note: DB2 UDB Version 7.2 is now provided with the OnDemand product package. Existing customers can obtain DB2 UDB Version 7.2 software on CD-ROM by ordering the OnDemand V7.1 Media Pack (see your IBM representative for more information). Customers can also upgrade to the functional equivalent of DB2 UDB Version 7.2 by applying Fix Pack 3 or later to DB2 UDB Version 7.1. The fix packs for DB2 UDB are available from IBM service on the Web at <ftp://service.software.ibm.com/ps/products/db2/fixes/>.

UNIX servers

This section lists the changes to DB2 information in *Installation and Configuration Guide for UNIX Servers*, GC27-0834-00.

Chapter 7. Installing the database manager. Installing DB2. Under **Installation notes**, number 1, delete the first bullet: *DB2 Client Application Enabler*. Beginning with DB2 UDB V7.1 Fix Pack 3, the DB2 Client Application Enabler is installed as part of the DB2 UDB Enterprise Edition option.

Windows servers

This section lists the changes to DB2 information in *Installation and Configuration Guide for Windows Servers*, GC27-0835-00.

1. Chapter 5. OnDemand system administrator account. This is the user account that you will use to install DB2. Grant the following user rights to the OnDemand system administrator account in Windows:
 - Act as part of the operating system
 - Create a token object
 - Increase quotas
 - Log on as a service
 - Replace a process level token
2. Chapter 7. Installing the database manager. Please ignore the section titled **Installing DB2**. To install DB2 on the library server, follow the instructions in *DB2 Universal Database for Windows V7 Quick Beginnings*, GC09-2971 to perform a typical installation and verify the installation.

Chapter 12. Generic indexer

This section contains corrections to the *Indexing Reference*, SC27-0842-00 for the generic indexer.

Note: The system does not delete the source files that are specified on the GROUP_FILENAME: parameters in the generic index file. The system only deletes IND, OUT, and RES files.

CODEPAGE: parameter

The CODEPAGE: parameter is required; you must specify a valid code page in the generic index file. For example, CODEPAGE:819.

GROUP_FILENAME: parameter

Note: The system does not delete the source files that are specified on the GROUP_FILENAME: parameters in the generic index file. The system only deletes IND, OUT, and RES files.

The GROUP_FILENAME: parameter must specify the full path name of the input file to process. For example:

```
GROUP_FILENAME:/tmp/ondemand/inputfiles/f1b0a1600.afp
```

The documentation for the GROUP_FILENAME: parameter contains several instances of the parameter name that is missing the _ (underscore) character. The documentation is incorrect; the underscore character is a required part of the parameter name.

Parameter syntax

The parameter names in the generic index file are case sensitive and must appear in uppercase. For example, GROUP_FIELD_NAME:account is valid, while group_field_name:account is not.

Chapter 13. Hardware and software requirements

This section lists the changes and additions to the hardware and software requirements listed in *Introduction and Planning Guide*, GC27-0839-00, *Installation and Configuration Guide for UNIX Servers*, GC27-0834-00, and *Installation and Configuration Guide for Windows Servers*, GC27-0835-00.

1. OnDemand supports the IBM @server pSeries™ and IBM @server xSeries™ product lines.
2. For Oracle users, OnDemand now requires Oracle 8i Release 3 (8.1.7.0) or later. See Chapter 19, “Oracle information,” on page 65 for more information about using Oracle with OnDemand.
3. OnDemand now requires Sun Solaris Version 8 or later on Sun servers.
4. IBM no longer distributes or supports the OS/2® client or supports the Windows client on Windows 95.
5. For the Windows client and the administrative client, OnDemand now requires Microsoft® Windows NT® Version 4.0 SP5 or later, Windows 98, Windows 2000, or Windows XP. **Note:** The AFP Viewer (which is used in the Windows client) does not support the Clear Type Font support that is available in Windows XP.
6. For Windows NT servers, OnDemand now requires Microsoft Windows NT Server Version 4.0 SP5 or later.
7. The OnDemand CICS/ESA client now requires OS/390 Version 2 Release 6 and CICS/ESA Version 4.1.0.
8. The OnDemand PDF Indexer feature is included in the OnDemand media pack. However, customers are not authorized to install or use the OnDemand PDF Indexer feature that is included in the OnDemand media pack unless they first purchase the appropriate Proofs of Entitlement for the feature.
9. To view PDF documents from the Windows client, IBM recommends that you use Adobe Acrobat or Adobe Acrobat Approval. You must purchase Adobe Acrobat and Adobe Acrobat Approval from Adobe. See Chapter 2, “Adobe software,” on page 11 for more information.
10. If you plan to use the report wizard or the graphical indexer to process PDF input files with the administrative client, then you must use Adobe Acrobat or Adobe Acrobat Approval. You must purchase Adobe Acrobat and Adobe Acrobat Approval from Adobe. See Chapter 2, “Adobe software,” on page 11 for more information.
11. If you are using the OnDemand Web Enablement Kit (ODWEK) to view PDF documents from Internet Explorer 5.5 with SP2 or later or Internet Explorer 6, you should update your Acrobat viewing software to Adobe Acrobat Version 5 or later. See Chapter 30, “Web Enablement Kit,” on page 127 for more information.
12. For AIX 4.3.3 or later servers, you must install the C Set ++ Runtime for AIX 4.3 (x1C.rte 5.0.0.0) before you install or upgrade to OnDemand Version 7.1. You can get the latest version of this software from IBM Software on the Web at:

`ftp.software.ibm.com/aix/products/ccpp/vacpp5_runtime.tar.Z`

13. If you upgrade AIX to Fix Pack 9, depending on your use of OnDemand, you may have intermittent problems with ARSSOCKD and ARSLOAD, causing you to restart the system or causing loads to fail. See Chapter 4, "AIX information," on page 15 for details.
14. To use the Send function of the OnDemand client with Lotus Notes, you must use Lotus Notes Version 5.0.8 or later. (The Send function provides an interface to a Mail Application Programming Interface (MAPI) client to send documents to people that do not have access to OnDemand.)
15. To use the Java API that is provided with the Web Enablement Kit (ODWEK) requires Java version 1.2.2 or later. **Note:** The Java API is set of APIs that reproduces the CGI interface for Java developers. The Java API can be used in place of the CGI interface.
16. The Java servlet that is provided with ODWEK requires Java 1.2.2 or later. **Note:** The Java servlet can be used in place of the CGI interface (or the Java API).
17. OnDemand Version 7.1.0.7 and later has been tested with the TSM Version 5.1 server, TSM Version 5.1 API, and TSM Version 5.1 Administrative Client on all platforms (AIX, HP-UX, Solaris, and Windows) and operates correctly.
18. Customers that plan to use Java version 1.3.1 to support the Java API or the Java Servlet that is provided with ODWEK must install Java version 1.3.1 with Fix Pack 4 or later.

Chapter 14. Installing on UNIX

This section provides the latest information on configuring OnDemand on UNIX servers to run under a user other than root.

Beginning with Version 2.2.1.10, new installations (instances) of OnDemand can be configured to run under a user other than the root user. After you install the OnDemand product, you should create a user for the OnDemand instance owner, set permissions for the cache storage file systems and the OnDemand files, give the instance owner permission to write to the system console, and configure the ARS.INI file. After you complete these steps, you can then create the OnDemand instance. See the *Installation and Configuration Guide for UNIX Servers* for information about installing software and creating the instance.

Note: This section describes how to configure OnDemand on a standard library / object server system (library server and object server on the same workstation or node). If you plan to run a distributed library / object server system, with one or more object servers on different workstations or nodes than the library server, then you should also configure OnDemand on the object servers.

Important: If you are using Oracle, then you can run OnDemand under a user other than the root user, so long as you store all application group data in the SYSTEM tablespace. (You should verify that all application groups have the Use Tablespace option set to No. See the Database Information page in application groups for information about the Use Tablespace option.)

Creating a user for the OnDemand instance owner

Note: You should have created a group when you installed the database manager software. The group has administrator authority for the database and the database file systems. We recommend that you add the user for the OnDemand instance owner to the database owner's group to give the OnDemand instance owner administrator authority for the database and the database file systems.

Run the `mkuser` (or `useradd`) command to create a user for the OnDemand instance owner. For example, in AIX, run the following command to make a new user:

```
mkuser id=1004 pgrp=dbagroup groups=dbagroup home=/home/userid
      userid passwd userid
```

where **1004** represents the user ID, **dbagroup** represents the database owner's group, **/home/userid** represents the user's home directory, **userid** represents the username and instance name, and **passwd userid** represents the user's initial password.

You should use a unique user for each instance that you create. This allows for easier error recovery if a system error occurs.

Setting permissions

You need to give the OnDemand instance owner the following authorities and permissions:

- Administrator authority for the database. You can do this by adding the OnDemand instance owner to the database owner's group.
- Ownership of the cache storage file systems that are listed in the ARS.CACHE file. You can do this by running the Change Owner command for each file system that is listed in the ARS.CACHE file and specifying the user and group for the OnDemand instance owner.
- Permission to read the OnDemand configuration files. Make sure that the OnDemand instance owner has permission to read the following files:
 - ARS.CACHE
 - ARS.CFG
 - ARS.DBFS
 - ARS.INI
- Permission to read and execute the OnDemand script files. Make sure that the OnDemand instance owner has permission to read and execute the following files:
 - ARSLOG
 - ARSPRT
- Permission to write to the console. Make sure that the OnDemand instance owner has permission to write to the system console.

Configuring the ARS.INI file

Add the following line to the ARS.INI file:

```
SRVR_INSTANCE_OWNER=userid
```

where **userid** represents the name of the OnDemand instance owner that you created in "Creating a user for the OnDemand instance owner" on page 45.

Configuring object servers

If you plan to run a distributed library / object server system, with one or more object servers on different workstations or nodes than the library server, then you should also configure OnDemand on each of the object servers.

To configure OnDemand on the object servers, do the following:

- Create a group and user for the OnDemand instance owner.
- Give ownership of the cache storage file systems listed in the ARS.CACHE file to the group and user for the OnDemand instance owner.
- Give permission to read the following files to the OnDemand instance owner:
 - ARS.CACHE
 - ARS.CFG
 - ARS.INI
- Give permission to write to the console to the OnDemand instance owner.
- Add the following line to the ARS.INI file:

```
SRVR_INSTANCE_OWNER=userid
```


where **userid** represents the name of the user for the OnDemand instance owner.

Chapter 15. Messages

Beginning with Version 7.1.0.14, message number ARS0090I (message number 90 in the system log) has been changed. The new message will list the number of documents that were retrieved. For example:

```
90 BulkProcess: Time(0.246) NumDocs(1123) Srvr->server.company.xyz.com 0.1.2.3<-
```

Chapter 16. Migrating from previous versions of OnDemand

If you are migrating your OnDemand system to Version 7.1, then you should backup your databases and other files that are critical to the operation of the system before you install OnDemand Version 7.1. See “Preparing databases for migration” and “Preparing archive storage for migration” on page 52 for more information.

Important: You can migrate directly from Version 2.2.1.0 or later to Version 7.1. If you are running a version earlier than Version 2.2.1.0, then:

- For UNIX servers, you must upgrade to Version 2.2.1.0 before you migrate to Version 7.1
- For Windows servers, you must first upgrade to Version 2.2.0.15 and then upgrade to Version 2.2.1.0 before you migrate to Version 7.1

For information about upgrading to Version 2.2.1.0, see the README file that is provided with Version 2.2.1.0. You can get the README file from IBM service on the Web at:

<ftp://service.software.ibm.com/software/ondemand/fixes/v221>

The enhancements included in Version 7.1 require changes to the OnDemand system tables (the database). “Migrating from a previous version” on page 52 describes the steps that you must complete to migrate from a previous version of OnDemand.

OnDemand has always provided backward compatibility between clients and servers. However, please note the following:

- The library server and the object servers must always use the same version of the product.
- The administrative client and the library server should always use the same version of the product.
- You cannot use an administrative client prior to Version 7.1 to administer a Version 7.1 server.
- When a new version of the client and the server contains new function, you should not use the new function unless you use the same (new) version of both the client and the server. For example, if you plan to use the new PNG data type, then you must use both the Version 7.1 or later client and the Version 7.1 or later server.

Preparing databases for migration

Follow the instructions in the information that is provided with the database management product that you are using with OnDemand to prepare the OnDemand database for migration. For example, if you are using DB2, see the *Quick Beginnings* publication for your library server operating system. See the section titled Migrating from a previous release of DB2.

In general, IBM recommends that you create a full offline backup of the database on removable media. No processes or users should be connected to the database, except for the backup task. Save the backup copy in a safe location. You will need the backup copy if you want to return to the previous version.

Preparing archive storage for migration

Follow the instructions in the information that is provided with the archive storage management product that you are using with OnDemand to prepare the archive storage manager database for migration. For example, if you are using TSM, see the *Quick Start* publication for the TSM server operating system. See the section titled Migrating from a previous version of TSM.

In general, IBM recommends that you create a full backup of the archive storage manager database on removable media. Save the backup copy in a safe location. You will need the backup copy if you want to return to the previous version.

You should also save copies of the archive storage manager configuration and operation files. For example, if you are using TSM, then save the following files: DSMSEV.DSM, DSMSEV.OPT, DSM.OPT, DSM.DB2.OPT, the device configuration file, the volume history file, and any scripts that you may have written. You should also save information about the archive storage devices that are attached to the system (for example, in AIX, save the output from `lsdev -Cc tape` and `lsdev -Cc library`).

Migrating from a previous version

This section describes how to migrate from Version 2.2.1 to Version 7.1. If you need to migrate more than one instance, then you must repeat the steps for each instance. If your OnDemand system contains servers on more than one physical node or workstation, then you must repeat the steps on each node or workstation.

1. Verify that the server meets all of the hardware, software, and memory requirements to install OnDemand. For more information, see Chapter 13, "Hardware and software requirements," on page 43 and the *Introduction and Planning Guide*, GC27-0839-00.
2. Backup databases. See "Preparing databases for migration" on page 51 and "Preparing archive storage for migration."
3. If you are migrating a UNIX server, then make a copy of the following files: ARS.CFG, ARS.INI, ARS.CACHE, ARS.DBFS, ARSLOAD.CFG, and ARS_ADSM and any user exit programs and scripts that you may have written. If you are migrating a Windows server, then you should backup the Registry and save any user exit programs that you may have written.
4. Make a backup copy of the following files: ARSLOG, ARSPRT.
5. If required, upgrade and configure the database manager software. See your database manager product information for details.
6. If required, upgrade and configure the archive storage manager software. See your archive storage manager product information for details.
7. If you are migrating a UNIX server, then run the ARSLINK program to delete the links to the Version 2.2.1 server programs. First, change to the OnDemand configuration file directory. (The configuration file directory is `/usr/lpp/ars/config` in AIX and `/opt/ondemand/config` in HP-UX and Sun Solaris.) Then run the following command:

```
arslink -u
```

Note: The ARSLINK program no longer exists in Version 7.1; the `bin/srvr` directory no longer exists and there are no longer symbolic links to the ARSADMIN, ARSMAINT, ARSOBJD, ARSSOCKD, and ARSTBLSP programs.

8. Remove any OnDemand Version 2.2.1 software from the server. The uninstall process that you use will vary, depending on the operating system of the server. However, you need to delete any prior versions of the client and the server software from the system before you continue.
9. Install the OnDemand Version 7.1 software. See the *Installation and Configuration Guide* for details.
10. If you are migrating a UNIX server, then configure the following files: ARS.CFG, ARS.INI, ARS.CACHE, ARS.DBFS, ARSLOAD.CFG, and ARS_ADSM. Use the information from the files that you saved in step 3 on page 52.
11. Reconfigure any scripts and recompile any user exit programs that you may have written for previous versions to work with Version 7.1. Use the information from the files that you saved in step 3 on page 52.
12. Configure the following files: ARSLOG, ARSPRT. Use the information from the files that you saved in step 4 on page 52.
13. Verify initialization processes and scheduled tasks. For example, you may need to modify `/etc/inittab` and `crontab`.
14. Upgrade the OnDemand system tables. Complete the following steps:
 - ___ Step a. Open a window with a command prompt.
 - ___ Step b. Create a temporary directory. Create the directory on a drive that has at least 100 MB of free space. (The exact amount of temporary space required during the migration will depend on the number of user-defined application groups and folders and the number of annotations that you have stored in the system.)
 - ___ Step c. Make the temporary directory the current directory.
 - ___ Step d. For Windows servers that use SQL Server, complete the following steps:
 - 1) Run the ARSDB program to drop the configuration indexes. First, start the OnDemand for WinNT Command Window. Then enter the following command at the prompt:


```
arsdb -ev
```
 - 2) Use the ISQL tool to alter the OnDemand tables. You can run the ISQL tool interactively and enter the SQL statements to alter the tables at the prompt or you can run the ISQL tool and specify the name of an input file that contains the SQL statements.

To enter the SQL statements at the prompt, first enter the ISQL command:

```
isql -E
```

Then, at the prompt, enter the SQL statements shown in Figure 1 on page 54. Replace the string `ondemand_database` in the USE statement with the name of the OnDemand database in SQL Server (for example, ARCHIVE). Replace the string `instance_owner` in each of the ALTER TABLE statements with the name of the instance owner of the OnDemand database in SQL Server (for example, ODADMIN).

```

USE ondemand_database

GO

ALTER TABLE instance_owner.arsag ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsag ADD upd_date BIGINT NULL
ALTER TABLE instance_owner.arsag ADD last_doc_date BIGINT NULL
ALTER TABLE instance_owner.arsag ADD migr_srvr_str VARCHAR(254) NULL

GO
ALTER TABLE instance_owner.arsann ADD table_name VARCHAR(18) NULL
ALTER TABLE instance_owner.arsann ADD doc_exp_date BIGINT NULL

GO
ALTER TABLE instance-owner.arsapp ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsapp ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsfol ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance-owner.arsfol ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsgroup ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsgroup ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsnode ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsnode ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsprt ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsprt ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsres ADD add_date BIGINT NULL

GO

```

Figure 1. Altering the OnDemand Tables in SQL Server (Part 1 of 2)


```

ALTER TABLE instance_owner.arsset ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsset ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arssys ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arssys ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsuser ADD email VARCHAR(254) NULL
ALTER TABLE instance_owner.arsuser ADD upd_userid VARCHAR(128) NULL
ALTER TABLE instance_owner.arsuser ADD upd_date BIGINT NULL

GO

ALTER TABLE instance_owner.arsann ALTER COLUMN time_stamp BIGINT NOT NULL

GO

ALTER TABLE instance_owner.arsfolfldusr ALTER COLUMN min_int BIGINT NOT NULL
ALTER TABLE instance_owner.arsfolfldusr ALTER COLUMN max_int BIGINT NOT NULL

GO

ALTER TABLE instance_owner.arsload ALTER COLUMN start BIGINT NOT NULL
ALTER TABLE instance_owner.arsload ALTER COLUMN stop BIGINT NOT NULL
ALTER TABLE instance_owner.arsload ALTER COLUMN exp_date BIGINT NOT NULL

GO

ALTER TABLE instance_owner.arsseg ALTER COLUMN start_date BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN stop_date BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN post_date BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN closed_date BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN reimported_date BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN last_update BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN last_backup BIGINT NOT NULL
ALTER TABLE instance_owner.arsseg ALTER COLUMN last_stats BIGINT NOT NULL

GO

ALTER TABLE instance_owner.arssys ALTER COLUMN time_out BIGINT NOT NULL

GO

ALTER TABLE instance_owner.arsuser ALTER COLUMN last_update BIGINT NOT NULL

GO

```

Figure 1. Altering the OnDemand Tables in SQL Server (Part 2 of 2)

To specify the name of a file that contains the SQL statements, enter:

```
isql.exe -E -i sqlserver.txt
```

Where `sqlserver.txt` is the name of the file that contains the SQL statements. See Figure 1 on page 54 and remember to replace the string `ondemand_database` in the `USE` statement with the name of the OnDemand database in SQL Server (for example, `ARCHIVE`) and replace the string `instance_owner` in

each of the ALTER TABLE statements with the name of the instance owner of the OnDemand database in SQL Server (for example, ODADMIN).

___ Step e. Export the OnDemand system tables using the arsdB program.

Note: In the steps that follow, replace the string `../ondemand/bin` with the name of the OnDemand program directory on the system:

AIX	<code>/usr/lpp/ars/bin</code>
HP-UX	<code>/opt/ondemand/bin</code>
Sun Solaris	<code>/opt/ondemand/bin</code>
Windows	<code>\Program Files\IBM\OnDemand for Windows NT\bin</code>

If the database manager is DB2 or SQL Server, then enter:

```
../ondemand/bin/arsdb -lxv
```

If the database manager is Oracle, then enter:

```
../ondemand/bin/arsdb -xv
```

___ Step f. Drop the old tables:

```
../ondemand/bin/arsdb -dv
```

___ Step g. Create the new tables and indexes:

```
../ondemand/bin/arsdb -rtv
```

___ Step h. Import the old table information into the new tables. If the database manager is DB2 or SQL Server, then enter:

```
../ondemand/bin/arsdb -ilv
```

If the database manager is Oracle, then enter:

```
../ondemand/bin/arsdb -iv
```

___ Step i. Last, run maintenance on the new tables. If the database manager is DB2 or Oracle, then enter:

```
/opt/ondemand/bin/arsdb -mv
```

If the database manager is Oracle, then enter:

```
../ondemand/bin/arsdb -sv
```

15. Verify your OnDemand installation. You may need to restart your system before you verify the installation.
16. Optionally, create a full offline backup image of the database.

Chapter 17. Naming rules

This section lists the changes to the rules for naming objects in OnDemand, found in the *Introduction and Planning Guide*, GC27-0839-00.

Removal of trailing blank characters: Beginning with Version 7.1.0.14, trailing blank characters are trimmed from the name when adding or updating an application, application group, folder, printer, storage set, distribution file, or distribution group. As a reminder, for existing objects, the name specified to a command line program must include trailing blank characters that are part of the name.

- User ID. Can contain one to 128 characters.

Note: For a local server, the default administrative userid is admin. The system does not set an initial password for the admin userid on a local server. See Chapter 40, "Local server setup," on page 171 for more information on local servers.

- Password. Can contain one to 128 characters.

Notes:

1. OnDemand security verifies only the first eight characters that are entered by the user. The additional characters are provided for customers who choose to implement their own password security by enabling the Security User Exit.
 2. If the user types more than eight characters, the client will display an error message unless the SRVR_FLAGS_IGNORE_PASSWD_MAX parameter in the ARS.INI file is set to 1 (one). The default value for the SRVR_FLAGS_IGNORE_PASSWD_MAX parameter is 0 (zero).
 3. If your organization enables the Security User Exit, you should set the Minimum Password Length option to Permit Blank Password so that OnDemand security does not validate passwords that are entered by your users (when they set or change a password). Also, OnDemand security ignores the Maximum Password Age option when you enable the Security User Exit.
 4. Unless your organization enables the Security User Exit, IBM recommends that you specify a value of 8 (eight) or less for the Minimum Password Length option.
- Group. Can contain one to 128 characters.
 - Server Queue. Can contain one to 60 characters.
 - Storage Node Logon. Can contain one to 128 characters.
 - Storage Node Password. Can contain one to 128 characters.

Chapter 18. National language support

Character data conversion tables

Note: The information in this section is for customers who plan to use the OnDemand Windows and administrative clients to view double-byte character set (DBCS) AFP data or DBCS or single-byte character set (SBCS) line data.

Introduction

The OnDemand clients automatically convert characters in a document from the code page of the server to the code page of the client by using ICONV converters and conversion tables. This method of character mapping supports DBCS languages (for AFP and line data) and SBCS languages (for line data), including most DBCS User Defined Character (UDC) mappings.

The ICONV converters automatically map the user-defined area of a code page to the standard user-defined area of the corresponding conversion table. If the code page contains UDC mappings outside of the standard user-defined area, you can create your own mappings and generate and use your own conversion table; otherwise, the clients will not be able to display the characters correctly. You should use the ICONVDEF program that is provided with OnDemand to generate your own conversion table.

This section provides an overview of the conversion process. Customers should contact the IBM Globalization Center of Competency (GCoC) for help with modifying conversion tables. A good source of information with details including pictorial examples is the *Character Data Representation Architecture Reference and Registry*, SC09-2190. Additionally, some newer table packages, such as those that support the new Chinese standard GB18030, contain specific documentation within the package file.

The following sections provide more information:

- Obtaining source conversion tables
- Modifying conversion tables
- Generating an updated conversion table
- Distributing an updated conversion table
- ICONVDEF program reference

Obtaining source conversion tables

IBM maintains a Character Data Reference Architecture (CDRA) conversion table repository at the IBM developerWorks Web site at <http://www.ibm.com/developerworks/toolbox>.

Note: You must register as a guest to gain access to the files on the Web site.

After you receive your subscription ID, return to the IBM developerWorks Web site. Then click on the Download Catalog link. From the Download Catalog page, click on Complete Catalog. In the table of contents, select Sample Code. Under

Sample Code, scroll down to the section titled NLS. Take the Character Data Conversion Tables link to the conversion tables.

Before you begin downloading files, read all of the information on the Character Data Conversion Tables page. Then download the `intro.zip` file. The `intro.zip` file contains a `readme.txt` file, a `hints.txt` file, and a `tabindex.txt` file.

- The `readme.txt` file contains important information that you should read first.
- The `hints.txt` file contains useful information about the code page values used for some special encodings. It will give you the code page value that has been used for UCS-2, it will give you the code page values for code pages containing the new Euro currency symbol and it may provide you with some useful information if you are looking for tables for non-IBM code pages.
- The `tabindex.txt` file has been set up to help you locate a table. Each record is made up of five columns. The first two columns contain the decimal values of the code page source and target. The lower value is always in the first column. For example, if you are looking for a conversion table for code page 850 to code page 500, you would look for the directory entry with 500 in column one and 850 in column two. The third column indicates the name of the conversion table ZIP file within the package ZIP file. Column four contains the name of the package ZIP file that contains the conversion table and should be downloaded. Column five contains a short, somewhat cryptic description of the conversion pair. For example, the conversion table for the source code page 937 (Traditional Chinese EBCDIC) to code page 13488 (Traditional Chinese UCS-2) is in the `03A934B0.zip` file that is contained in the `package2.zip` file.

Note: The ICONV converters used by OnDemand convert from the source code page to UCS-2 and UCS-2 to the target code page. All conversion tables must convert from the source and target code page to UCS-2. This means that you can only use conversion table files in which the last four characters are 34B0 (the target code page, 13488 in hexadecimal).

Modifying ICONV tables

Customers should contact the IBM Globalization Center of Competency (GCoC) for help with modifying conversion tables. You can contact the GCoC by email at gcoc@ca.ibm.com or by mail to:

IBM Canada Limited
Globalization Center of Competency
8200 Warden Avenue
Markham, Ontario
L6G 1C7
Canada

Be sure to include one of the following in the subject line of the note:

- Comments about the CDRA Conversion Table Repository
- Help with Modifying a Character Data Conversion Table

Note: The ICONV converters used by OnDemand convert from the source code page to UCS-2 and UCS-2 to the target code page. All conversion tables must convert from the source and target code page to UCS-2. This means that you can only use conversion table files in which the last four characters are 34B0 (the target code page, 13488 in hexadecimal) and the file type is

UPMAP110. For example, 03A934B0.upmap110 is the name of the conversion table file used to generate a UCS-2 conversion table for Traditional Chinese code page 937.

Generating an updated ICONV table

You should use the ICONVDEF program to generate a conversion table that contains your modifications. The ICONVDEF program reads a source translation file from a specified input file and writes the compiled version to a specified output file. See “ICONVDEF program reference” for more information and an example of using the ICONVDEF program.

Distributing an updated conversion table

After generating a new conversion table, you need to distribute the table to your users so that the OnDemand client will have access to the table when a user views an AFP or line data document that requires the character conversion. The easiest way to distribute a new table to many users is to add the table to the client installation files. That way, when a user runs the installation program to install or update the client, the table is automatically copied to the client workstation along with the other OnDemand client files.

See the section titled “Distributing User-Defined Files” in the *Windows Client Customization Guide* for requirements and instructions about adding a table to the OnDemand client installation files.

Note: Conversion table files (for example, the IBM-937 file) must be located in the \program files\ibm\ondemand32\locale\uconvtab directory on the client. You should add any conversion table files that you modify to the \Ars32\Custom\Locale\uconvtab directory on the distribution server so that the client installation program can copy them into the correct installation directory (by default, the \program files\ibm\ondemand32\locale\uconvtab directory).

ICONVDEF program reference

The ICONVDEF program reads a source conversion file from a specified input file and writes the compiled conversion table to a specified output file. If you do not specify an input file or you do not specify an output file, ICONVDEF uses standard input (stdin) and standard output (stdout), respectively. The source conversion file contains directives that are acted upon by the ICONVDEF utility to produce the compiled version of the conversion table.

The ICONVDEF program is provided with the OnDemand server software in the \program files\ibm\ondemand for winnt\bin directory.

Note: You must use the ICONVDEF program that is distributed with the OnDemand for Windows server to compile conversion tables for the Windows client. Conversion tables that are compiled on other platforms (for example, AIX) are not compatible with the Windows operating system.

The syntax of the ICONVDEF program is:

```
►► —i convdef — [ -f —sourceConversionFile ] [ -v ] —outputConversionTable— ►►
```

Where:

-f specifies the name of the *sourceConversionFile* to process. If you do not specify an input file, ICONVDEF uses standard input (stdin). The source conversion file contains directives that are acted upon by the ICONVDEF utility to produce the compiled version of the conversion table.

-v requests more verbose information. Echoes input to stdout.

outputConversionTable specifies the name of the output file to contain the compiled version of the conversion table. If you do not specify an output file, ICONVDEF uses standard output.

The following example shows how to use the ICONVDEF program to generate a UCS-2 conversion table for Traditional Chinese code page 937.

```
iconvdef -f 03A934B0.upmap110 IBM-937
```

Where:

03A934B0.upmap110 in the name of the input conversion file that contains the user-defined mappings. This file was extracted from the 03A934B0.ZIP file within the PACKAGE2.ZIP file from the CDRA conversion table repository.

IBM-937 is the name of the output conversion table. This file must be copied to the \program files\ibm\ondemand32\locale\uconvtab directory on the client.

Language codes in UNIX

Table 23 on page 168 of the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Installation and Configuration Guide for UNIX Servers*, GC27-0834-00 contains several errors. Table 2 lists the correct code set for the CHS language code in HP-UX and the correct locale codes for the CHT language code in AIX and HP-UX.

Table 2. Supported Language Codes, Locales, and Code Sets

Language Code	Territory	Operating System	Locale	Code Set
CHS	China (PRC)	HP-UX	zh_CN.hp15CN	hp15cn
CHT	Taiwan	AIX	zh_TW	IBM-eucTW
		HP-UX	zh_TW.eucTW	eucTW

DBCS code pages

Added the **code page** to the third column of Table 24 on page 170 in the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Installation and Configuration Guide for UNIX Servers*, GC27-0834-00. Table 3 shows the updated table.

Table 3. DBCS code pages

Territory	Host Code Page	UNIX Code Page (Code Set)	PC Code Page
China (PRC)	IBM-935	IBM-1383 (IBM_eucCN)	IBM-1386 (MS-936)
Japan	IBM-939	IBM-954 (IBM_eucJP)	IBM-943 (MS-932)
Korea, South	IBM-933	IBM-970 (IBM_eucKR)	IBM-1363 (MS-949)
Taiwan	IBM-937	IBM-964 (IBM_eucTW)	IBM-950 (MS-950)

Mapping AFP fonts

IBM provides font definition files for the following multi-byte languages:

- Japanese – CODED.JPN and CSDEF.JPN
- Korean – CODED.KOR and CSDEF.KOR
- Simplified Chinese – CODED.CHS and CSDEF.CHS
- Traditional Chinese – CODED.CHT and CSDEF.CHT

For example, for Japanese, if a font is listed in the CSDEF.JPN file, it will be supported in the AFP viewer. The actual font mapping from AFP font to Windows font happens in the [FGID] section of the CSDEF.JPN file (the [FGID] section is at the end of the file). If a font is not listed in the CSDEF.JPN file, you can add the font by editing the CSDEF.JPN file with a standard text editor.

See the *Windows Client Customization Guide* for details about how to map AFP fonts.

Traditional Chinese clients

Beginning with Version 7.1.0.8, the Traditional Chinese client may incorrectly display the contents of the Description field on the Open a Folder dialog box when logged on to a local server.

Chapter 19. Oracle information

This section provides additional information about using Oracle with OnDemand. This section also includes corrections to the Oracle information that is contained in the OnDemand product publications. Please make a note of these items, which supersede the information in the OnDemand product publications.

Support for Oracle 9i

Beginning with Version 7.1.0.14, OnDemand now supports Oracle 9i. See the *Oracle 9i Database Installation Guide* for your operating system for information about system requirements, installation instructions, and other information.

OnDemand has been tested with Oracle 9i on the following operating systems:

Operating System	Version
AIX	AIX 4.3.3 with maintenance level 9
HP-UX	HP-UX 11i with December 2001 Goldbase and Goldapps 11i
Solaris	Solaris 8 with recommended and security patch set
Windows	Windows 2000 Server Windows Server 2003 Enterprise Edition

Installation notes for UNIX servers

The ARS_ORACLE_HOME parameter in the ARS.CFG file must be set to the directory in which Oracle was installed. See Chapter 5, "ARS.CFG," on page 17 for more information.

On page 49 of the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Installation and Configuration Guide for UNIX Servers*, GC27-0834, replace the section titled **Integrate OnDemand with the Oracle Shared Library** with the following:

You must do the following so that the OnDemand programs can run with Oracle:

AIX Either create a link to the shared library in the /usr/lib directory:

```
cd /usr/lib
ln -s oracleDir/lib32/liblntsh.a .
```

Where oracleDir is the name of the directory in which Oracle is installed.

Or set the following variable:

```
LIBPATH=oracleDir/lib32
```

Where oracleDir is the name of the directory in which Oracle is installed.

HP-UX

Either create a link to the shared library in the /usr/lib directory:

```
cd /usr/lib
ln -s oracleDir/lib32/libc1ntsh.sl.9.0 libc1ntsh.sl.8.0
```

Where oracleDir is the name of the directory in which Oracle is installed.

Or set the following variable:

```
SHLIB_PATH=oracleDir/lib32
```

Where oracleDir is the name of the directory in which Oracle is installed.

Solaris

Either create a link to the shared library in the /usr/lib directory:

```
cd /usr/lib
ln -s oracleDir/lib32/libcIntsh.so.9.0 libcIntsh.so.8.0
```

Where oracleDir is the name of the directory in which Oracle is installed.

Or set the following variable:

```
LD_LIBRARY_PATH=oracleDir/lib32
```

Where oracleDir is the name of the directory in which Oracle is installed.

Field types and field sizes

Table 4 lists the types of database fields supported by OnDemand and the number of bytes required to hold a value in each type of field. For decimal fields, the actual field size will vary, depending on the average precision for each column. IBM recommends that you read your Oracle information about the NUMBER datatype.

Table 4. Database Field Types and Sizes

OnDemand Field Type	SQL Type	Oracle Datatype	Length
Small Integer	SQL_SMALLINT	NUMBER	7 bytes
Integer	SQL_INT	NUMBER	11 bytes
Decimal	SQL_FLOAT	NUMBER	4 – 22 bytes (maximum precision is 32) ¹
String (Fixed)	SQL_CHAR	CHAR	1 – 254; 1 byte per character declared, even if partially used
String (Variable)	SQL_VARCHAR	VARCHAR2	1 – 254; 1 byte per character plus 4 bytes overhead; unused characters do not consume storage
Date	SQL_SMALLINT	NUMBER	7 bytes
Time	SQL_INT	NUMBER	11 bytes
Date/Time	SQL_INT	NUMBER	11 bytes

1. Oracle stores decimal values in a variable length format. Each value is stored in scientific notation, with one byte used to store the exponent and up to twenty bytes to store the mantissa. The resulting value is limited to 32 digits of precision. Oracle does not store leading and trailing zeros. For example, the number 412.50 is stored in a format similar to 4.125 * 100, with one byte used to store the exponent (2) and two bytes used to store the four significant digits of the mantissa (4, 1, 2, 5). Taking this into account, the column data size for a particular decimal value NUMBER (p), where p is the precision of a given value (scale has no effect), can be calculated using the formula: 1 + FLOOR(p/2) + 2. Therefore, the system requires a minimum of four bytes to hold a decimal value.

Table 4. Database Field Types and Sizes (continued)

OnDemand Field Type	SQL Type	Oracle Datatype	Length
Date/Time (TZ)	SQL_INT	NUMBER	11 bytes

Estimating database storage space

Estimating the size of database objects is an imprecise undertaking. Overhead caused by disk fragmentation, free space, and the use of variable length fields (including numbers) make size estimation difficult, because there is such a wide range of possibilities for field types and row lengths. After initially estimating your database size, you should create a test database and populate it with representative data.

Figure 2 shows a formula that you can use to estimate disk storage space for the database. You should use the formula to estimate the amount of database storage space for each of the reports that you plan to load on the system. You can use the formula for reports that contain logical items, such as statements and policies, and for reports that contain sorted transaction data.

Note: The formula was derived in part from information provided by Oracle. For more information, or if you have special requirements or if you need to do more, see the Oracle product information. Also, the formula does not include space requirements related to file management overhead required by the operating system, including file block size and directory control space.

$$\begin{aligned}
 \text{TableSize} &= (\text{Sum of column lengths}) + 3 + (\text{Number of columns} * 2) \\
 \text{IndexSize} &= (\text{Index 1 length} + 8) + (\text{Index 2 length} + 8) + \dots \\
 \text{DatabaseSize} &= ((\text{TableSize} + 40) * 1.2) + (\text{IndexSize} * 1.2) \\
 &\quad * \text{Number of indexed items per month} \\
 &\quad * \text{Number of months to keep index in database}
 \end{aligned}$$

Figure 2. Formula to Estimate Database Storage Space

- Index n length is the size of a database field for which you want OnDemand to build an index. For example, an integer field requires eleven bytes to hold the index value. Oracle requires an additional eight bytes for each index that you define.
- OnDemand adds approximately 40 bytes of control information to each row in a table.
- The formula includes a 20 percent buffer for overhead.
- When the report contains logical items, the Number of indexed items per month is the number of statements, policies, and so forth.
- When the report contains sorted transaction data, the Number of indexed items per month is the number of groups of indexed pages. By default, the system indexes a report in groups of 100 pages. You can specify the size of an indexed group of pages when you index a report with ACIF.

The following example shows how to estimate the database storage space required for a report that contains logical items, such as statements. The example is for indexing one million items per month and keeping the index data in the database for 24 months. Table 5 on page 68 lists the database fields used in the example.

Approximately 3.75 GB of disk space is required to maintain the index data in the database.

$$\begin{aligned} \text{TableSize} &= (59 + 3 + (4 * 2)) = 70 \\ \text{IndexSize} &= (12 + 8) = 20 \\ \text{DatabaseSize} &= ((70 + 40) * 1.2) + (20 * 1.2) = 156 \\ &\quad * 1000000 = 156000000 \\ &\quad * 24 = 3744000000 \end{aligned}$$

Figure 3. Estimating Database Storage Space — Part I

Table 5. Estimating Database Storage Space — Part II

Field Name	Field Type	Field Size	Index or Filter
Report Date	Date	7 bytes	Filter
Account Number	Fixed String	12 bytes	Index
Invoice Balance	Decimal	9 bytes	Filter
Customer Name	Fixed String	31 bytes	Filter

Estimating the size of rollback segments

The database storage space requirements for your OnDemand system include a rollback segment, an area on your disk subsystem that is used when OnDemand makes changes to the database. The size of the rollback segment should be based on the types of transactions that run against the database.

In general, there are two types of transactions in OnDemand: loading data into the system and deleting data from the system. Loading data into the system is a batch job of (usually) long running transactions. The OnDemand actions that load data into the system include the ARSLOAD program and the ARSADMIN LOAD command. Deleting data from the system can be either a long running transaction (for example, when deleting an entire report from the system) or a short transaction (for example, when deleting a document from the system). The OnDemand actions that delete data from the system include the ARSADMIN UNLOAD command, the ARSDOC DELETE command, and the ARSMAINT program. Data is deleted by the ARSMAINT program when you set an expiration type in your application groups.

Because of this mix of transaction sizes, most customers should plan a large rollback segment that can handle transactions of any size. Most customers should plan to allocate enough storage space for a rollback segment that can hold the transactions for the largest input file that will be loaded into the system.

To estimate the size of the rollback segment, you need to consider three factors:

1. The number of documents that are loaded into the system during the single largest load process.
2. The number of bytes that are allocated to the user-defined database fields for the application group that is associated with the single largest load process.
3. The 40 bytes of system information that OnDemand adds to each database row.

Once you know these values, you can use the formula shown in Figure 4 on page 69 to estimate the size of the rollback segment that is required for your system.

Note: The formula was derived in part from information provided by Oracle. For more information, or if you have special requirements or if you need to do more, see the Oracle product information. Also, the formula does not include space requirements related to file management overhead required by the operating system, including file block size and directory control space.

$$\begin{aligned} & (\text{Number of documents in largest load} * \\ & \quad (\text{Number of bytes in application group} + 40 \text{ OnDemand overhead bytes})) \\ & \quad * 2 = \text{estimated size of rollback segment} \end{aligned}$$

Figure 4. Formula for Estimating the Size of the Rollback Segment

For example, suppose that your largest OnDemand load is for a statement application that loads 150,000 statements in a single load file. The OnDemand application group database fields require approximately 50 bytes of database storage per document. Figure 5 shows the example calculation, which requires approximately 27 MB of rollback segment space.

$$\begin{aligned} & (150,000 \text{ statements} * \\ & \quad (50 \text{ bytes in application group} + 40 \text{ OnDemand overhead bytes})) \\ & \quad * 2 = 27,000,000 \end{aligned}$$

Figure 5. Example of Estimating the Size of the Rollback Segment

If you expect to delete data from the system by using the ARSDOC DELETE command (this is a somewhat unusual requirement), then instead of using the size of the single largest load file, you should substitute the largest number of records that you expect to delete during a delete process.

Tuning the database

In general, most customers use OnDemand in one of two ways:

- Long-term archive for larger reports. These customers create table spaces for each application group that they add to the system, load many rows into the database at one time, maintain data on the system for many months or years, and delete a report at a time from the system. These parameters usually result in very static data and a low maintenance operation.
 - A table space contains data from one application group.
 - Inserts are done by a high-volume batch process.
 - After a table reaches its Maximum Rows value, OnDemand closes the table and no additional inserts are made to the table. Closed tables remain available for queries until the data is removed from the system.
 - Data is removed from the system by dropping a table or deleting a large number of consecutive rows at a time.

Because of the low rate of change within the tables, these customers should seldom or never need to tune their database.

- Short-term archive for smaller reports. These customers store all application group data in the SYSTEM table space (or in one or more DATA table spaces), load very few rows into the database at one time, maintain data on the system for a short period of time, or use the Delete Document method to remove data from the system. These parameters usually result in very dynamic data and a high maintenance operation.
 - A table space contains data from more than one application group.
 - Inserts are done by a low-volume batch process.

- Inserts and deletes happen frequently.
- Data is removed from the system by deleting one row at a time.

Because of the high rate of change within the tables, these customers should plan to tune their database on a time-based schedule, such as a weekly or nightly process. **Note:** Some customers tune their database every day. However, most customers do not tune their database until the optimizer ignores the current set of statistics or generates an inefficient plan. IBM recommends that these customers tune the database just before they plan to take an offline backup of the database. This schedule keeps the optimization information up to date and minimizes the impact to system availability, because a system outage is already planned.

Tuning the database is done by collecting statistics on the tables, which can provide faster access to the data, thereby improving performance. Statistics on tables are gathered by using the ANALYZE command. When you analyze a table, its associated indexes are automatically analyzed as well. The frequency with which you analyze the tables depends on the rate of change within the tables. **Note:** If you collect statistics and do not notice a visible performance improvement, then dropping and recreating the indexes to your tables may help. Customers in a high maintenance operation may need to periodically rebuild the indexes on their most active tables. See your Oracle information for details about rebuilding indexes.

OnDemand provides two programs to collect statistics on database tables: the ARSDB program and the ARSMAINT program.

Using the ARSDB program

You can use the ARSDB program to collect statistics on the OnDemand system tables, such as the user table, the group table, the application group table, and so forth. For most customers, the OnDemand system tables require very little maintenance. You can probably schedule the ARSDB program to collect statistics once a month (or less often).

The syntax is:

```
/opt/ondemand/bin/arsdb <options>
```

The options are:

- e Drop configuration indexes
- r Create configuration indexes
- s Collect statistics

Using the ARSMAINT program

You can use the ARSMAINT program to maintain the tables that contain user-defined application group data. User-defined application groups are the application groups that you define to the system. Customers in a high maintenance operation should run the ARSMAINT program on a regular schedule.

The syntax is:

```
/opt/ondemand/bin/arsmaint <options>
```

The options are:

- d, –i Expire index data from the database. The –i parameter expires index data

that has been imported from archive storage. If you do not migrate index data to archive storage, then you do not need to specify the `-i` parameter.

`-e` Migrate index data from the database to archive storage. If you do not migrate index data to archive storage, then you do not need to specify the `-e` parameter.

`-r` Collect statistics. **Note:** The ARSMaint program collects statistics only on the tables that have changed since the last time that statistics were collected. OnDemand keeps information about all of its tables, including the last time that it modified a table and the last time that it collected statistics on a table.

`-g applGroup`

Process the tables for the specified application group. If you do not specify this parameter and name an application group, then the ARSMaint program processes all of the user-defined application groups.

Chapter 20. Password security

Beginning with Version 7.1.0.7, the OnDemand security function for userids has been changed. The new function requires that the password be changed:

- When a user accesses the system for the first time
- When a System Administrator or a User Administrator changes or resets a user's password

In addition, for customers that are upgrading to Version 7.1.0.7 or later, any existing user who has never changed their password will be prompted to change the password the first time that they log on to the system after the upgrade is completed.

Note: This change does not affect customers that use the Security User Exit to implement their own user / password security (and bypass the OnDemand security function).

A password in OnDemand can contain one to 128 characters.

- OnDemand security verifies only the first eight characters that are entered by the user. The additional characters are provided for customers who choose to implement their own password security by enabling the Security User Exit.
- If the user types more than eight characters, the client will display an error message unless the `SRVR_FLAGS_IGNORE_PASSWD_MAX` parameter in the `ARS.INI` file is set to 1 (one). The default value for the `SRVR_FLAGS_IGNORE_PASSWD_MAX` parameter is 0 (zero).
- If your organization enables the Security User Exit, you should set the Minimum Password Length option to Permit Blank Password so that OnDemand security does not validate passwords that are entered by your users (when they set or change a password). Also, OnDemand security ignores the Maximum Password Age option when you enable the Security User Exit.
- Unless your organization enables the Security User Exit, IBM recommends that you specify a value of 8 (eight) or less for the Minimum Password Length option.

Chapter 21. PDF Indexer

General information

Note: The OnDemand PDF Indexer feature is included in the OnDemand media pack. However, customers are not authorized to install or use the OnDemand PDF Indexer feature that is included in the OnDemand media pack unless they first purchase the appropriate Proofs of Entitlement for the feature.

The PDF Indexer now supports DBCS languages. However, IBM does not provide any DBCS fonts. You can purchase DBCS fonts from Adobe. IBM supports all DBCS fonts, except for the encrypted Japanese fonts. See “Specifying the location of Adobe fonts” for important information about configuring the system so that the PDF Indexer can locate DBCS font files that may be required to process a PDF input file.

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 workstation on which you are running the PDF Indexer.

Change to the FIELD parameter. Beginning with Version 7.1.0.4, a field that is based on the location of a trigger (BASE=TRIGGER) can be defined at any location on the page that contains the trigger. Previously, a field that was based on the location of a trigger had to be defined to the right and below the upper left point of the trigger. With this change, the x or y values can be negative, so long as the resulting absolute field coordinates of the field string rectangle are still in the range of $0 \leq x \leq 45$ and $0 \leq y \leq 45$. The $ul(x,y)$ and $lr(x,y)$ coordinates of the FIELD parameter are relative offsets from the $ul(x,y)$ coordinates of the trigger. For example, suppose the field string rectangle is located at $ul(1,1)$, $lr(2,2)$ which is an absolute location on the page. If the trigger string rectangle is located at $ul(5,5)$, $lr(7,7)$, then the field coordinates would be $ul(-4,-4)$, $lr(-3,-3)$.

Change to the MASK option of the FIELD parameter. Beginning with Version 7.1.0.4, the string that you can specify for the mask can contain any character. Previously, the string that you could specify for the mask could contain only the valid mask symbols (the @, #, -, ^, %, and = characters).

Specifying the location of Adobe 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 will substitute 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 DBCS fonts, verify the locations of the DBCS font files and export or add the ACRO_RES_DIR and PSRESOURCEPATH environment variables.

The standard Adobe fonts should always be installed in the standard font directory on the system. Table 6 on page 76 shows the default locations.

Table 6. Location of Standard Adobe Font Files

Platform	Standard Font Directory
AIX	/usr/lpp/ Acrobat3/Fonts
HP-UX	/opt/ Acrobat3/Fonts
Solaris	/opt/ Acrobat3/Fonts
Windows	\WINNT\Fonts

DBCS font files must be installed in the directories specified in Table 7 so that the PDF indexer can access DBCS fonts that may be required to process a PDF input file.

Table 7. Location of Adobe DBCS Font Files

Platform	Font Directory	CMap File Directory	CIDFont File Directory
AIX	/usr/lpp/ Acrobat3/Fonts	/usr/lpp/ Acrobat3/Fonts/Resource/CMap	/usr/lpp/ Acrobat3/Fonts/Resource/CIDFont
HP-UX	/opt/ Acrobat3/Fonts	/opt/ Acrobat3/Fonts/Resource/CMap	/opt/ Acrobat3/Fonts/Resource/CIDFont
Solaris	/opt/ Acrobat3/Fonts	/opt/ Acrobat3/Fonts/Resource/CMap	/opt/ Acrobat3/Fonts/Resource/CIDFont
Windows	\WINNT\Fonts	\WINNT\Fonts\ Resource\CMap	\WINNT\Fonts\ Resource\CIDFont

The environment variables listed in Table 8 must be exported or added so that the PDF indexer can locate DBCS font files.

Table 8. Environment Variables for Adobe DBCS Font Files

Environment Variable	Purpose
ACRO_RES_DIR	Location of the CMap files
PSRESOURCEPATH	Locations of the CIDFont, CMap, and DBCS font files

For AIX:

```
export ACRO_RES_DIR=/usr/lpp/ Acrobat3/Fonts/Resource/CMap
export PSRESOURCEPATH=/usr/lpp/ Acrobat3/Fonts:/usr/lpp/ Acrobat3/Fonts/Resource/
CMap:/usr/lpp/ Acrobat3/Fonts/Resource/CIDFont
```

For HP-UX:

```
export ACRO_RES_DIR=/opt/ Acrobat3/Fonts/Resource/CMap
export PSRESOURCEPATH=/opt/ Acrobat3/Fonts:/opt/ Acrobat3/Fonts/Resource/
CMap:/opt/ Acrobat3/Fonts/Resource/CIDFont
```

For Solaris:

```
export ACRO_RES_DIR=/opt/ Acrobat3/Fonts/Resource/CMap
export PSRESOURCEPATH=/opt/ Acrobat3/Fonts:/opt/ Acrobat3/Fonts/Resource/
CMap:/opt/ Acrobat3/Fonts/Resource/CIDFont
```

For Windows:

Add the System Variables listed in Table 9 on page 77 to the Environment Variables

dialog box.

Table 9. System Environment Variables for Adobe DBCS Font Files

Variable	Value
ACRO_RES_DIR	\WINNT\Fonts\Resource\CMap
PSRESOURCEPATH	\WINNT\Fonts;\WINNT\Fonts\Resource\CMap;\WINNT\Fonts\Resource\CIDFont

TRACEDD parameter

The TRACEDD parameter is new in Version 7.1.0.13. For more information, see “Trace facility.”

Trace facility

Beginning with Version 7.1.0.13, an enhanced tracing capability for the PDF indexer is now available. The tracing capability provides assistance to users attempting to debug problems, such as when the system fails during the indexing and loading of PDF documents.

To trace or debug a problem with the PDF indexer, the following is required:

- The parameter file, which specifies the fields, triggers, indexes and other indexing information
- The PDF input file to process

The parameter file and PDF input file can be processed by running the PDF indexer from the command line. For example:

```
arspdoci parmdd=filen.parms inputdd=filen.pdf outputdd=filen.out indexdd=filen.ind  
tracedd=filen.trace
```

Where:

arspdoci is the name of the command-line version of the PDF indexer program

parmdd= specifies the name of the input file that contains the indexing parameters

inputdd= specifies the name of the PDF input file to process

outputdd= specifies the name of the output file that contains the indexed PDF documents created by the PDF indexer

indexdd= specifies the name of the output file that contains the index information that will be loaded into the database

tracedd= specifies the name of the output file that contains the trace information

Note: See the *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Indexing Reference*, SC27-0842 for more information about the parameters that may be specified when running the ARSPDOCI program.

After running the PDF indexer with the trace, the output file specified by the tracedd= parameter will contain detailed information about the processing that took place and where the PDF indexer is failing during the process. The trace information will identify whether a trigger was not found, a field was not found, the PDF data is corrupted, there was a problem extracting a PDF page from the document, or even if there is not enough memory or disk space to complete the

required operations. Figure 6 shows an example of the trace information that may be generated by the PDF indexer.

```

COORDINATES=IN
ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
TRIGGER1=UL(7.00,0.25),LR(7.70,0.57),*, 'Page: '
ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
ARSPDOCI completed code parse_trigger <-----
ARSPDOCI completed code parse_quoted_parm <-----
ARSPDOCI completed code parse_quoted_parm 001 ----->
ARSPDOCI completed code parse_trigger 001 ----->
FIELD1=UL(7.00,0.48),LR(7.90,0.77),0,(TRIGGER=1,BASE=0)
ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
ARSPDOCI completed code parse_field <-----
ARSPDOCI completed code parse_subfields <-----
ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
ARSPDOCI completed code parse_subfields 001 ----->
ARSPDOCI completed code parse_field 001 ----->
FIELD2=UL(6.11,1.39),LR(7.15,1.57),0,(TRIGGER=1,BASE=0)
ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
ARSPDOCI completed code parse_field <-----
ARSPDOCI completed code parse_subfields <-----
ARSPDOCI completed code get_keyword <-----

.
.
.

ARSPDOCI completed code get_keyword <-----
ARSPDOCI completed code get_keyword 003 ----->
ARSPDOCI completed code arspparm_final_sanity_check <-----
ARSPDOCI completed code arspparm_final_sanity_check 001 ----->
ARSPDOCI completed code ArspProcessOpt <-----
ARSPDOCI completed code ArspOpenIndex <-----
ARSPDOCI completed code ArspOpenIndex 001 ----->
Adobe PDF Library version -732512488.-1
Editing is : -1
Number of input pages = 130
ARSPDOCI completed code ArspProcessOpt:Calling ArspSearchDocPages()
ARSPDOCI completed code ArspSearchDocPages <-----
ARSPDOCI completed code ArspSearchDocPages: ArspCreateWordFinder()
ARSPDOCI completed code ArspSearchDocPages: PDWordFinderAcquireWordList()
ARSPDOCI completed code ArspSearchDocPages: PDDocAcquirePage()
ARSPDOCI completed code ArspSearchDocPages: ArspSearchPage()
ARSPDOCI completed code ArspSearchDocPages: PDPageRelease()
ARSPDOCI completed code ArspSearchDocPages: PDWordFinderReleaseWordList()
Trigger(s) not found by page 1
ARSPDOCI completed code ArspSearchDocPages 004 ----->
ARSPDOCI completed code ArspProcessOpt:Calling ArspCloseIndex()
ARSPDOCI completed code ArspCloseIndex <-----
ARSPDOCI completed code ArspCloseIndex 001 ----->
ARSPDOCI completed code ArspProcessOpt:Calling PDDocClose()
ARSPDOCI completed code ArspProcessOpt 002 ----->
ARSPDOCI completed code 1
ARSPDOCI completed code ArspFreeParms ()

```

Figure 6. Trace Information for the PDF Indexer

Processing PDF input files with the graphical indexer

This section describes how to use the graphical indexer to create indexing information for PDF input files.

Important: If you plan to use the report wizard or the graphical indexer to process PDF input files, then you must first install Adobe Acrobat or Adobe Acrobat Approval on the PC from which you plan to run the administrative client. You must purchase Adobe Acrobat and Adobe Acrobat Approval from Adobe.

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 will copy the 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. 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`, where `x.y` is the version of Acrobat, for example, 4.0, 5.0, and so forth.

Beginning with Version 7.1, you can define indexing information in a visual environment. You begin by opening a sample input file with the graphical indexer. You can run the graphical indexer from the report wizard or by choosing the sample data option from the Indexing Information page of the application. After you open an input file in the graphical indexer, you define triggers, fields, and indexes. The PDF indexer uses the triggers, fields, and indexes to locate the beginning of a document in the input data and extract index values from the input data. Once you have defined the triggers, fields, and indexes, you can save them in the application so that OnDemand can use them later on to process the input files that you load into the system.

You define a trigger, field, or index by drawing a box around a text string with the mouse and then specifying properties. For example, to define a trigger that identifies the beginning of a document, you could draw a box around the text string Account Number on the first page of a statement in the input file. Then, on the Add a Trigger dialog box, you would accept the default values provided, such as the location of the text string on the page. When processing an input file, the PDF indexer attempts to locate the specified string in the specified location. When a match occurs, the PDF indexer knows that it has found the beginning of a document. The fields and indexes are based on the location of the trigger.

The PDF file that you open with the graphical indexer should contain a representative sample of the type of input data that you plan to load into the system. For example, the sample input file must contain at least one document. A good sample should contain several documents so that you can verify the location of the triggers, fields, and indexes on more than one document. The sample input file must contain the information that you need to identify the beginning of a document in the input file. The sample input file should also contain the information that you need to define the indexes. When you load an input file into the system, the PDF indexer will use the indexing information that you create to locate and extract index values for each document in the input file.

The following 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.

1. To begin, start the administrative client.
2. Log on to a server.
3. Start the report wizard by clicking the Report Wizard icon on the toolbar. The report wizard opens the Sample Data dialog box.
4. Click Select Sample Data to open the Open dialog box.
5. Type the name or full path name of a file in the space provided or use the Look in or Browse commands to locate a file.
6. Click Open. The graphical indexer opens the input file in the report window.
7. Press F1 to open the main help topic for the report window. The main help topic contains general information about the report window and contains 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).
8. Close any open help topics and return to the report window.
9. Define a trigger.
 - Find a text string that uniquely identifies the beginning of a document. For example, Account Number, Invoice Number, Customer Name, and so forth.
 - Using the mouse, draw a box around the text string. Start just outside of the upper left corner of the string. Click and hold mouse button one. 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 enclosed the text string completely inside of a box, release the mouse button. The graphical indexer highlights the text string inside of a box.
 - Click the Define a Trigger icon on the toolbar to open the Add a Trigger dialog box. Verify the attributes of the trigger. For example, the text string that you selected in the report window should be displayed under Value; for Trigger1, the Pages to Search should be set to Every Page. Click Help for assistance with the other options and values that you can specify.
 - Click OK to define the trigger.
 - To verify that the trigger uniquely identifies the beginning of a document, first put the report window in display mode. Then click the Select tool to open the Select dialog box. 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. Use the Select dialog box to move forward to the first page of each document and return to the first document in the input file.
 - Put the report window in add mode.
10. Define a field and an index.
 - 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.
 - Using the mouse, draw a box around the text string. Start just outside of the upper left corner of the string. Click and hold mouse button one. 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 enclosed the text string completely inside of a box, release the mouse button. The graphical indexer highlights the text string inside of a box.

- Click the Define a Field icon on the toolbar to open the Add a Field dialog box.
 - On the Field Information page, 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.
 - On the Database Field Attributes page, verify the attributes of the database field. In the Database Field Name space, enter the name of the application group field into which you want OnDemand to store the index value. In the Folder Field Name space, enter the name of the folder field that will appear on the client search screen. Click Help for assistance with the other options and values that you can specify.
 - Click OK to define the field and index.
 - To verify the locations of the fields, first put the report window in display mode. The fields should have a blue box drawn around them. Next, click the Select tool to open the Select dialog box. 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. Use the Select dialog box to move forward to each document and display the field. Then return to the first document in the input file.
 - Put the report window in add mode.
11. Click the Display Indexer Parameters tool to open the Display Indexer Parameters dialog box. The Display Indexer Parameters dialog box lists the indexing parameters that the PDF indexer will use to process the input files that you load into the application. At a minimum, you need one trigger, one field, and one index. See the *Indexing Reference* for details about the indexing parameters.
 12. When you have finished defining all of the triggers, fields, and indexes, close the report window.
 13. Click Yes to save the changes to the indexer parameters.
 14. On the Sample Data window, click Next to continue with the report wizard.

Chapter 22. Publications

April 2003: An updated version of *IBM Content Manager OnDemand for Multiplatforms Version 7.1: Web Enablement Kit Implementation Guide*, SC27-1000-02 is now available. You can obtain a copy of the *Web Enablement Kit Implementation Guide* publication directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

March 2003: An updated version of *IBM Content Manager OnDemand Version 7.1: Messages and Codes*, SC27-1379-02 is now available. You can obtain a copy of the *Messages and Codes* publication directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

An updated version of *IBM Content Manager OnDemand Version 7.1: User's Guide*, SC27-0836-01 is now available. You can obtain a copy of the *User's Guide* directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

An updated version of *IBM Content Manager OnDemand Version 7.1: Windows Client Customization Guide*, SC27-0837-01 is now available. You can obtain a copy of the *Windows Client Customization Guide* directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

Chapter 23. Publications CD-ROM

The OnDemand publications CD-ROM contains the OnDemand books in HTML and PDF formats. The OnDemand books on the CD-ROM are provided in several languages.

CD-ROM directory structure

The HTML files on the CD-ROM have an extension of HTM. The PDF files on the CD-ROM have an extension of PDF. The files are located in one of the following directories. On UNIX systems:

```
/cdrom/<language>/html/<book>  
/cdrom/<language>/pdf
```

On Windows systems:

```
x:\<language>\html\<book>  
x:\<language>\pdf
```

Where:

/cdrom

Refers to your mount point (UNIX)

x: Refers to your CD-ROM drive (Windows)

<language>

Refers to the language code, and is named with one of the codes listed in Table 10.

Table 10. Directory Language Identifier

Directory	Language	Directory	Language
ARA	Arabic	FRA	French
CHS	Simplified Chinese	FRC	Canadian French
CHT	Traditional Chinese	JPN	Japanese
DEU	German	KOR	Korean
ENU	English	NLD	Dutch
ESP	Spanish	PTB	Brazilian Portuguese

<book>

Refers to the HTML directory for each book, and named using the five-character identifier assigned to each book (see Table 11 on page 86).

Notes:

1. The directory names may appear in uppercase or lowercase, depending on your operating system.
2. Not all of the OnDemand books that are viewable from the CD-ROM are translated into all of the supported languages. If a particular book is translated into a particular language, then it is in the language directory specific to that language. This means, for example, that the English language books are in the \enu directory on Windows, and in the /enu directory on UNIX, whereas the

French books are in the \fra directory on Windows, and in the /fra directory on UNIX. Whenever a book is not available in a specific language, the English book is provided.

OnDemand publications

Table 11 shows the OnDemand books that are on the CD-ROM and the directory name (HTML) for each publication.

Table 11. OnDemand Publications

Title	Directory Name
Introduction and Planning Guide	ars1p
Installation Guide for UNIX Servers	ars1u
Installation Guide for Windows Servers	ars1w
Administrator's Guide	ars1b
Indexing Reference	ars1d
User's Guide	ars5q
Windows Client Customization Guide	ars5u
Web Enablement Kit Installation and Configuration Guide	ars5y

Note: Not all of the books are translated into all of the supported languages. Whenever a book is not available in a specific language, the English book is provided.

Viewing the HTML books

The books included with OnDemand are in HTML soft copy format. The HTML format enables you to browse the information and provides hypertext links to related information. It also makes it easier to share the books across your organization.

You can view the HTML books with any browser that conforms to HTML Version 3.2 specifications.

To view the HTML books:

1. Insert the OnDemand publications CD-ROM into the CD-ROM drive. On UNIX systems, mount the CD-ROM. See your operating system documentation for the mounting procedures.
2. Start the browser.
3. Click File->Open and open the desired book from one of the following locations:
 - On UNIX systems:
`/cdrom/%L/HTML/<book>` directory on the CD-ROM, where `/cdrom` represents the mount point of the CD-ROM, `%L` represents the three-character country / region code that represents your language (for example, ENU for English), and `<book>` represents the directory name. See Table 10 on page 85 for a list of the codes and Table 11 for a list of directory names.
 - On Windows systems:

x:\language\HTML\<>book> directory, where x represents the CD-ROM drive, language represents the three-character country / region code that represents your language (for example, ENU for English), and <book> represents the directory name. See Table 10 on page 85 for a list of the codes and Table 11 on page 86 for a list of directory names.

4. The main HTML file for each book is INDEX.HTM. Load this file into your browser to view the book.

You can also copy the HTML files from the CD-ROM to a local or network drive and browse them from there.

Printing the PDF books

If you prefer to have printed copies of the books, you can print the PDF files found on the OnDemand publications CD-ROM. Using the Adobe Reader, you can print either the entire book or a specific range of pages. For the file name of each book on the CD-ROM, see Table 11 on page 86.

You can obtain the latest version of Adobe Reader from Adobe on the Web at:

<http://www.adobe.com>

The PDF files are included on the OnDemand publications CD-ROM with a file extension of PDF. To access the PDF files:

1. Insert the OnDemand publications CD-ROM into the CD-ROM drive. On UNIX systems, mount the CD-ROM. See your operating system documentation for the mounting procedures.
2. Start the Adobe Reader.
3. Open the desired PDF file from one of the following locations:
 - On UNIX systems:
/cdrom/doc/%L/PDF directory on the CD-ROM, where /cdrom represents the mount point of the CD-ROM and %L represents the three-character country / region code that represents your language (for example, ENU for English). See Table 10 on page 85 for a list of the codes.
 - On Windows systems:
x:\doc\language\PDF directory, where x represents the CD-ROM drive and language represents the three-character country / region code that represents your language (for example, ENU for English). See Table 10 on page 85 for a list of the codes.

You can also copy the PDF files from the CD-ROM to a local or network drive and print them from there.

Ordering the printed books

You can order the printed OnDemand books from IBM Publications on the Web at:

<http://www.ibm.com/shop/publications/order>

Chapter 24. Remote library server

Beginning with version 7105, you can now use the Windows server configurator program to add an instance of OnDemand that is running on some other system. This feature supports using the IBM Content Manager CommonStore product on a Windows system to access an OnDemand library server that is running on an AIX, HP-UX, iSeries™, Solaris, or z/OS system or some other Windows system. To configure the system to use a remote library server, you must install the OnDemand software on the system that is running the CommonStore software, add a Local Windows Server with the OnDemand configurator program, and configure an instance of the Local Windows Server to access the remote library server.

To configure an instance to access a remote library server, you must specify the TCP/IP host name alias, fully-qualified host name, or IP address of the system on which the remote library server is running and you must specify the TCP/IP port number of the remote library server. The programs on the local Windows system communicate with the remote library server through the specified TCP/IP port.

This feature also supports running the OnDemand server programs (such as ARSDB, ARSDOC, and ARSLOAD) from the command line on a Windows system to access an instance of OnDemand that is running on an AIX, HP-UX, iSeries, Solaris, or z/OS system or some other Windows system. This scenario requires that you install the OnDemand software on the local Windows system, add a Local Windows Server with the OnDemand configurator program, and configure an instance of the Local Windows Server to access the remote library server. When you run the programs, you must specify the **-h** and **-I** parameters to identify the host and the instance that you want to process.

Chapter 25. Retrieval preview user exit

The retrieval preview user exit point allows for the modification of document data prior to the data being retrieved from the server. The retrieval preview user exit point allows you to run a user-written program to process documents that belong to the selected application. The user-written program is activated by selecting the Use Preview Exit option on the Miscellaneous Options page of an application. When the option is selected, the user-written program will be called any time that a request is made to retrieve a document. Any information that is specified in the Parameters field will be passed to the user-written program.

Notes:

1. The user-written program may be used to process single document retrievals only; the bulk data retrieval method is not supported, regardless of the type of client that is making the request, such as the Windows client, arsdoc get -n, and so on.
2. The retrieval preview user exit point is not currently available for iSeries or z/OS servers.
3. If a request is made to retrieve a large object document, care should be taken to make certain that the user-written program does not remove any pages from the document. The large object segment size and page navigation information are based on the number of pages that existed when the document was loaded on the server. Unexpected results may occur if this information is changed.
4. The retrieval preview user exit point may be enabled for all data types, except for None.
5. The user-written program can be used to remove sensitive information from a document or to perform other types of data manipulation. However, because the user-written program is not used during server reprinting or bulk retrievals, the restricted data may still be accessible by the user.
6. When modifying the data, the format and type of the data must not be changed; only the content may be changed. For example:
 - If the format of the data is EBCDIC data with a fixed length of 133 bytes, the format must not be changed to something different, such as ASCII data delimited by X'0A'
 - If the data type is AFP, the document may not be converted to some other type of data, such as PDF

When the modified data is viewed by the Windows client, the format of the data and the data type that is defined in the application on the View Information page will be used to display the data. If the format or data type has changed, the document will not view properly.

7. The retrieval user exit point may 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.
8. A sample user exit program arsuprep.c and header file arscsxit.h are provided by IBM. The files are located in the /usr/lpp/ars/exits directory

(AIX), /opt/ondemand/exits directory (HP-UX, Solaris), \program files\ibm\ondemand for winnt\exits (Windows).

Chapter 26. Support

Support for the OnDemand product is available from the Web. Click **Support** from the product Web site at:

<http://www.ibm.com/software/data/ondemand/mp>

The IBM support center maintains product updates for OnDemand. You can obtain the latest product updates from IBM service on the Web at:

<ftp://service.software.ibm.com/software/ondemand/fixes/v71>

If you encounter problems or errors running the OnDemand programs, you can call the IBM support center to obtain software program defect support. The phone number for the IBM support center is 1-800-237-5511. The OnDemand program ID is 5697-G34. The component ID for the OnDemand for Multiplatforms Version 7.1 server is 5697G3400. The component ID for the OnDemand Version 7.1 clients is 5697G3401.

Note: IBM is pleased to announce a consolidation of its Hardware (1-800-IBM-SERV) and Software (1-800-237-5511) support access numbers. Effective November 1, 2002, 1-800-IBM-SERV (1-800-426-7378) has been designated as the single point of contact number for both hardware and software support calls.

In addition to this consolidation, access to software support has been significantly simplified. Specifically, this new number streamlines the support process by reducing the numerous software support options to two. To help ensure a smooth transition while encouraging the use of 1-800-IBM-SERV for support calls, the current toll-free number for IBM's Software Support, 1-800-237-5511, will remain active with a notification message. The 1-800-237-5511 number will be discontinued within twelve months when it is anticipated that ample transition will have occurred.

In our effort to better serve our customers, IBM invites you to contact any of the following for additional information:

- Your IBM representative
- Your IBM Business Partner
- Call 1-800-IBM-4YOU (426-4968)

Chapter 27. System control tables

This section contains a description of each OnDemand system table, including column names and data types. This section also contains a description of the application group data tables (also known as dynamic tables).

The system tables are created when the OnDemand database is created by using the ARSDB program. The system tables are updated during normal operation in response to dynamic SQL statements.

Warning: The OnDemand system tables should not be modified or queried directly. Doing so may effect the operation or performance of the system. The system tables are subject to change in future versions of OnDemand.

List of tables

- "Application group table (ARSAG)" on page 96
- "Field mapping table (ARSAG2FOL)" on page 97
- "Application group field table (ARSAGFLD)" on page 98
- "Application group field alias table (ARSAGFLDALIAS)" on page 99
- "Application group permissions table (ARSAGPERMS)" on page 100
- "Annotation table (ARSANN)" on page 101
- "Application table (ARSAPP)" on page 102
- "User logical views table (ARSAPPUSR)" on page 103
- "Folder table (ARSFOL)" on page 104
- "Folder field table (ARSFOLFLD)" on page 105
- "Folder user fields table (ARSFOLFLDUSR)" on page 106
- "Folder permissions table (ARSFOLPERMS)" on page 107
- "Group table (ARSGROUP)" on page 108
- "Load table (ARSLOAD)" on page 109
- "Named query table (ARSNAMEQ)" on page 110
- "Node table (ARSNODE)" on page 111
- "Printer table (ARSPRT)" on page 112
- "Printer options table (ARSPRTOPTS)" on page 113
- "Printer user table (ARSPRTUSR)" on page 114
- "Resources table (ARSRES)" on page 115
- "Segment table (ARSSEG)" on page 116
- "Storage set table (ARSSET)" on page 117
- "System parameters table (ARSSYS)" on page 118
- "User table (ARSUSER)" on page 119
- "Users in groups table (ARSUSRGRP)" on page 120
- "User group ID table (ARSUSRGRPID)" on page 121
- "Application group data table" on page 122

Application group table (ARSAG)

This table maintains a catalog of application groups.

Table 12. Application group table (ARSAG)

Column Name	Data Type	Description
name	VARCHAR(60)	Application group name
description	VARCHAR(120)	Application group description
agid	INTEGER	Application group identifier
agid_name	VARCHAR(8)	Application group internal identifier
db_seg	CHAR(1) FOR BIT DATA	Segment type (Database Organization)
ann_type	CHAR(1) FOR BIT DATA	Annotation update
db_exp_date	INTEGER	Expire database indexes
db_mgrt_date	INTEGER	Migrate database indexes
sm_cache_doc	INTEGER	Storage manager cache definition
resgrp	INTEGER	Resource group identifier
load_id	INTEGER	Load Id counter
log	INTEGER	Log mask
load_id_suffix	VARCHAR(8)	Load Id suffix
mgrt_cache_date	INTEGER	Copy cache to archive storage
reimport_exp_date	INTEGER	Number of days to keep imported table before re-expiring
sid	INTEGER	Storage set identifier
expire_type	CHAR(1) FOR BIT DATA	Expiration type
seg_rows	INTEGER	Rows per segment (Maximum Rows)
seg_id	INTEGER	Segment Id counter
obj_size	INTEGER	Size of storage object
type	SMALLINT	Type of application group (Data Management; Create Tablespace Type)
upd_userid	VARCHAR(128)	Last user to update the application group
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time application group was last updated
last_doc_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Migrate last document date
migr_srvr_str	VARCHAR(254)	Migrate server string

Indexes

Unique Index	<ol style="list-style-type: none"> 1. name 2. agid
--------------	--

Field mapping table (ARSAG2FOL)

This table maintains a catalog of mappings between application group fields and folder fields. This table also contains a catalog of applications within application groups that can be queried from a folder.

Table 13. Field mapping table (ARSAG2FOL)

Column Name	Data Type	Description
fid	INTEGER	Folder identifier
agid	INTEGER	Application group identifier
aid	INTEGER	Application within the application group
folder_field	SMALLINT	Folder field identifier
appgrp_field1	SMALLINT	Application group field identifier
appgrp_field2	SMALLINT	Application group field identifier

Indexes

Unique Index (Oracle, SQL Server)	fid, agid, aid, folder_field
Unique Cluster Index (DB2)	fid, agid, aid, folder_field
Columns Included with Index (DB2 only)	appgrp_field1, appgrp_field2

Application group field table (ARSAGFLD)

This table maintains a catalog of application group fields.

Table 14. Application group field table (ARSAGFLD)

Column Name	Data Type	Description
agid	INTEGER	Application group identifier
field	SMALLINT	Application group field identifier
name	VARCHAR(18)	Application group field name
type	CHAR(1) FOR BIT DATA	Application group field type
qual	CHAR(1) FOR BIT DATA	Application group field type qualifier
mask	SMALLINT	Application group field mask
string_type	CHAR(1) FOR BIT DATA	Type of string field
string_len	SMALLINT	Length of string field
string_case	CHAR(1) FOR BIT DATA	Case of string field
leading	VARCHAR(8)	Leading characters to trim from string field
strip	VARCHAR(8)	Any characters to trim from string field
Trailing	VARCHAR(8)	Trailing characters to trim from string field

Indexes

Unique Index (Oracle, SQL Server)	agid, field
Unique Cluster Index (DB2)	agid, field
Columns Included with Index (DB2 only)	name, type, qual, mask, string_type, string_len, string_case, leading, strip, trailing

Application group field alias table (ARSAGFLDALIAS)

This table maintains a catalog of application group field aliases (Database values and Displayed values).

Table 15. Application group field alias table (ARSAGFLDALIAS)

Column Name	Data Type	Description
agid	INTEGER	Application group identifier
field	SMALLINT	Application group field identifier
internal	VARCHAR(254)	Internal (Database) value
external	VARCHAR(254)	External (Displayed) value

Indexes

Unique Index (Oracle, SQL Server)	agid, field
Unique Cluster Index (DB2)	agid, field

Application group permissions table (ARSAGPERMS)

This table maintains permissions specifications for application groups. Each row contains two primary elements: who (id) and what (agid) and specifies the user's authority for the application group, the user's document and annotation permissions, and a query restriction, if any.

Table 16. Application group permissions table (ARSAGPERMS)

Column Name	Data Type	Description
id	INTEGER	User identifier
agid	INTEGER	Application group identifier
id_perms	SMALLINT	User authority for the application group
doc_perms	SMALLINT	Document permissions
ann_perms	SMALLINT	Annotation permissions
q_restrict	LONG VARCHAR	Query restriction; user-defined and validated SQL string

Indexes

Unique Index (Oracle, SQL Server)	agid, id
Unique Cluster Index (DB2)	agid, id

Annotation table (ARSANN)

This table maintains a catalog of annotations.

Table 17. Annotation table (ARSANN)

Column Name	Data Type	Description
id	INTEGER	User identifier
userid	VARCHAR(128)	User name
agid	INTEGER	Application group identifier
doc_name	VARCHAR(11)	Document name
doc_off	INTEGER	Document offset
doc_len	INTEGER	Document length
comp_off	INTEGER	Compression offset
comp_len	INTEGER	Compression length
time_stamp	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Time annotation was created
page	INTEGER	Page (within document) on which annotation exists
type	INTEGER	Type of annotation
x_off	INTEGER	x offset on page
y_off	INTEGER	y offset on page
reserved	VARCHAR(32) FOR BIT DATA	(first byte is annotation color)
text_buf	LONG VARCHAR	The text of the annotation
image_buf	BLOB(1048576)	Any images that are attached to the annotation
table_name	VARCHAR(18)	Segment in which annotation exists
doc_exp_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Expiration date of document

Indexes

Index	id, agid, doc_name, doc_off, doc_len, comp_off, comp_len
-------	--

Application table (ARSAPP)

This table maintains a catalog of applications.

Table 18. Application table (ARSAPP)

Column Name	Data Type	Description
name	VARCHAR(60)	Application name
description	VARCHAR(120)	Application description
agid	INTEGER	Application group identifier
aid	INTEGER	Application identifier
doc_type	CHAR(1) FOR BIT DATA	Document type
doc_comp_type	CHAR(1) FOR BIT DATA	Default document compression type
res_comp_type	CHAR(1) FOR BIT DATA	Default resource compression type
idx_type	CHAR(1) FOR BIT DATA	Indexer
comp_obj_size	INTEGER	Compressed object size
pid	INTEGER	Default printer identifier
alias	VARCHAR(254)	Application identifier within application group
fixed_view	LONG VARCHAR	Default and fixed logical view information
indexer	LONG VARCHAR	Indexing parameters
preprocessor	LONG VARCHAR	Parameters used to preprocess rows before loading
upd_userid	VARCHAR(128)	Userid of last user to update application
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time application was last updated

Indexes

Unique Index (Oracle, SQL Server)	agid, aid, name
Unique Cluster Index (DB2)	agid, aid, name

User logical views table (ARSAPPUSR)

This table maintains a catalog of logical views.

Table 19. User logical views table (ARSAPPUSR)

Column Name	Data Type	Description
agid	INTEGER	Application group identifier
aid	INTEGER	Application identifier
id	INTEGER	User identifier
user_view	LONG VARCHAR	Logical view information

Indexes

Unique Index (Oracle, SQL Server)	agid, aid, id
Unique Cluster Index (DB2)	agid, aid, id

Folder table (ARSFOL)

This table maintains a catalog of folders.

Table 20. Folder table (ARSFOL)

Column Name	Data Type	Description
name	VARCHAR(60)	Folder name
description	VARCHAR(120)	Folder description
fid	INTEGER	Folder identifier
ann_search	CHAR(1) FOR BIT DATA	Annotation display
upd_userid	VARCHAR(128)	Userid of last user to update folder
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time folder was last updated

Indexes

Unique Index (Oracle, SQL Server)	name
Unique Cluster Index (DB2)	fid
Columns Included with Index (DB2 only)	name, description, ann_search, upd_userid, upd_date

Folder field table (ARSFOLFLD)

This table maintains a catalog of folder fields.

Table 21. Folder field table (ARSFOLFLD)

Column Name	Data Type	Description
fid	INTEGER	Folder identifier
field	SMALLINT	Folder field identifier
name	VARCHAR(60)	Folder field name
description	VARCHAR(120)	Folder field description
type	CHAR(1) FOR BIT DATA	Folder field type
qual	CHAR(1) FOR BIT DATA	Folder field type qualifier
map_type	SMALLINT	Type of field mapping

Indexes

Unique Index (Oracle, SQL Server)	fid, field
Unique Cluster Index (DB2)	fid, field
Columns Included with Index (DB2 only)	name, description, type, qual, map_type

Folder user fields table (ARSFOLFLDUSR)

This table maintains a catalog of folder fields for users that have specific field information for a folder.

Table 22. Folder user fields table (ARSFOLFLDUSR)

Column Name	Data Type	Description
id	INTEGER	User identifier
fid	INTEGER	Folder identifier
field	SMALLINT	Folder field identifier
query_order	SMALLINT	Order of fields in search criteria
dspl_order	SMALLINT	Order of fields in document list
sort_order	SMALLINT	Default sort order
sort_type	CHAR(1) FOR BIT DATA	Type of sort
usr_mask	SMALLINT	Field setup
ops	SMALLINT	Search operators
def_op	SMALLINT	Default search operator
def_value1	VARCHAR(254)	Default search value
def_value2	VARCHAR(254)	Default search value for Between, Not Between
min_int	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Smallest integer allowed
max_int	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Largest integer allowed
datefmt	VARCHAR(60)	Format of date strings in search criteria
datedef_type	CHAR(1) FOR BIT DATA	Default date units for interval
datedef_arg	INTEGER	Default date value for interval
min_float	FLOAT	Smallest decimal allowed
max_float	FLOAT	Largest decimal allowed
decimal_pt_pos	SMALLINT	Precision (right of decimal point)
string_wild	CHAR(1) FOR BIT DATA	Wildcard
dspl_datefmt	VARCHAR(60)	Format of date strings in document list

Indexes

Unique Index (Oracle, SQL Server)	fid, id, field
Unique Cluster Index (DB2)	fid, id, field
Columns Included with Index (DB2 only)	query_order, dspl_order, sort_order, sort_type, usr_mask, ops, def_op

Folder permissions table (ARSFOLPERMS)

This table maintains permissions specifications for folders. Each row contains two primary elements: the user (`id`) and the folder (`fid`) and specifies the user's authority for the folder, fields, and named queries.

Table 23. Folder permissions table (ARSFOLPERMS)

Column Name	Data Type	Description
<code>id</code>	INTEGER	User identifier
<code>fid</code>	INTEGER	Application group identifier
<code>perms</code>	SMALLINT	User permissions for the folder
<code>hits</code>	INTEGER	Maximum hits returned from a query

Indexes

Unique Index (Oracle, SQL Server)	<code>fid, id</code>
Unique Cluster Index (DB2)	<code>fid, id</code>
Columns Included with Index (DB2 only)	<code>perms, hits</code>

Group table (ARSGROUP)

This table maintains a catalog of groups.

Table 24. Group table (ARSGROUP)

Column Name	Data Type	Description
gid	INTEGER	Group identifier
id	INTEGER	User identifier
name	VARCHAR(128)	Name of group
description	VARCHAR(120)	Group description
upd_userid	VARCHAR(128)	Userid of last user to update group
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time group was last updated

Indexes

Unique Index	1. gid 2. name
--------------	-------------------

Load table (ARSLOAD)

This table maintains a record of all successful load processes.

Table 25. Load table (ARSLOAD)

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 load
start	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Start date in segment
stop	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Stop date in segment
exp_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Expiration date

Indexes

Unique Index	agid, exp_date
--------------	----------------

Named query table (ARSNAMEQ)

This table maintains a catalog of named queries.

Table 26. Named query table (ARSNAMEQ)

Column Name	Data Type	Description
fid	INTEGER	Folder identifier
id	INTEGER	User identifier
name	VARCHAR(60)	Name of named query
buf	LONG VARCHAR	Named query

Indexes

Unique Index (Oracle, SQL Server)	fid, id, name
Unique Cluster Index (DB2)	fid, id, name
Columns Included with Index (DB2 only)	name, description, logon, passwd, server, status, sec_nid, udp_userid, upd_date

Node table (ARSNODE)

This table maintains a catalog of storage nodes.

Table 27. Node table (ARSNODE)

Column Name	Data Type	Description
sid	INTEGER	Storage set identifier
nid	SMALLINT	Storage node identifier
name	VARCHAR(60)	Name of storage node
description	VARCHAR(120)	Description of storage node
logon	VARCHAR(128)	Storage manager logon userid
passwd	VARCHAR(128)	Storage manager logon password
server	VARCHAR(254)	Name of OnDemand object server
status	SMALLINT	Current location of storage set
sec_nid	SMALLINT	Secondary storage node
upd_userid	VARCHAR(128)	Userid of last user to update storage node
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time storage node was last updated

Indexes

Unique Index (Oracle, SQL Server)	sid, nid
Unique Cluster Index (DB2)	sid, nid

Printer table (ARSPRT)

This table maintains a catalog of printers.

Table 28. Printer table (ARSPRT)

Column Name	Data Type	Description
pid	INTEGER	Printer identifier
name	VARCHAR(60)	Name of printer
description	VARCHAR(120)	Description of printer
queue	VARCHAR(60)	Name of queue that is associated with the printer
type	CHAR(1) FOR BIT DATA	Type of printer
upd_userid	VARCHAR(128)	Userid of last user to update printer
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time printer was last updated

Indexes

Unique Index	1. name 2. pid
--------------	-------------------

Printer options table (ARSPRTOPTS)

This table maintains a catalog of printing options (parameters) for printers.

Table 29. Printer options table (ARSPRTOPTS)

Column Name	Data Type	Description
id	INTEGER	Printer identifier
name	VARCHAR(60)	Name of printer options
type	CHAR(1) FOR BIT DATA	Type of printer
buf	LONG VARCHAR	Printing options

Indexes

Unique Index (Oracle, SQL Server)	id, name
Unique Cluster Index (DB2)	id, name

Printer user table (ARSPRTUSR)

This table maintains a catalog of user permissions for printers.

Table 30. Printer user table (ARSPRTUSR)

Column Name	Data Type	Description
pid	INTEGER	Printer identifier
id	INTEGER	User identifier

Indexes

Unique Index (Oracle, SQL Server)	pid, id
Unique Cluster Index (DB2)	pid, id

Resources table (ARSRES)

This table maintains a catalog of AFP resources.

Table 31. Resources table (ARSRES)

Column Name	Data Type	Description
rid	INTEGER	Resource identifier
agid	INTEGER	Application group identifier
pri_nid	SMALLINT	Primary storage node identifier
sec_nid	SMALLINT	Secondary storage node identifier
comp_type	CHAR(1) FOR BIT DATA	Method of compression
add_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date that the resources was loaded into the system

Indexes

Unique Index (Oracle, SQL Server)	agid, rid
Unique Cluster Index (DB2)	agid, rid

Segment table (ARSSEG)

This table maintains a catalog of segments of application group data. There is one row for each segment of application group data.

The segments of application group data are also known as dynamic tables that are created by OnDemand.

Table 32. Segment table (ARSSEG)

Column Name	Data Type	Description
agid	INTEGER	Application group identifier
table_name	VARCHAR(18)	Application group segment table name
start_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Segment start date
stop_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Segment stop date
post_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date table was created
closed_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date table was closed
reimported_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date table was imported to the database
last_update	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date table was last updated
last_backup	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date of last table backup
last_stats	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date statistics last collected
mask	INTEGER	Location
ins_rows	INTEGER	Inserted rows
upd_rows	INTEGER	Updated rows
del_rows	INTEGER	Deleted rows
mod_rows	INTEGER	Modified rows
max_rows	INTEGER	Maximum number of rows

Indexes

Index	1. agid, start_date, stop_date 2. agid, closed_date
Unique Cluster Index (DB2)	agid, table_name

Storage set table (ARSSET)

This table maintains a catalog of storage sets.

Table 33. Storage set table (ARSSET)

Column Name	Data Type	Description
sid	INTEGER	Storage set identifier
name	VARCHAR(60)	Name of storage set
description	VARCHAR(120)	Description of storage set
load_type	CHAR(1) FOR BIT DATA	Type of load
upd_userid	VARCHAR(128)	Userid of last user to update storage set
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time storage set was last updated

Indexes

Unique Index (Oracle, SQL Server)	name
Unique Cluster Index (DB2)	sid
Columns Included with Index (DB2 only)	name, description, load_type, upd_userid, upd_date

System parameters table (ARSSYS)

This table contains a single row that specifies the current system parameters.

Table 34. System parameters table (ARSSYS)

Column Name	Data Type	Description
id	INTEGER	Identifier counter
name	VARCHAR(8)	Internal application group identifier
cdid	INTEGER	CD-ROM identifier
passwd_exp	INTEGER	Password expiration value
time_out	BIGINT (DB2, SQL Server) INTEGER (Oracle)	System inactivity timeout value
syslog_mask	INTEGER	Message logging options
exitlog_mask	INTEGER	User exit logging options
passwd_min_len	SMALLINT	Minimum password length
nid	SMALLINT	Internal storage node identifier
sys_mask	INTEGER	Case of userids and passwords
buf_resv	LONG VARCHAR	reserved
upd_userid	VARCHAR(128)	Userid of last user to update system parameters
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time system parameters were last updated

Indexes

The system parameters table is not indexed

User table (ARSUSER)

This table maintains a catalog of users.

Table 35. User table (ARSUSER)

Column Name	Data Type	Description
uid	INTEGER	User identifier
userid	VARCHAR(128)	User name
passwd	VARCHAR(128)	User's password
description	VARCHAR(120)	Description of user
admin	SMALLINT	User type and authority
last_update	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date password last set
pid	INTEGER	Default printer identifier
acct_info	VARCHAR(60)	User-defined accounting information
name	VARCHAR(60)	User's name
company	VARCHAR(60)	User's organization
title	VARCHAR(60)	User's title
addr1	VARCHAR(60)	User's address
addr2	VARCHAR(60)	User's address
addr3	VARCHAR(60)	User's address
addr4	VARCHAR(60)	User's address
dept	VARCHAR(60)	User's department
building	VARCHAR(60)	User's building or location
room	VARCHAR(60)	User's office or room
phone	VARCHAR(32)	User's phone number
fax	VARCHAR(32)	User's fax number
cover_page	VARCHAR(60)	Default fax cover page
time_out	BIGINT (DB2, SQL Server) INTEGER (Oracle)	User's inactivity timeout value
email	VARCHAR(254)	User's e-mail address
upd_userid	VARCHAR(128)	Userid of last user to update user
upd_date	BIGINT (DB2, SQL Server) INTEGER (Oracle)	Date and time user was last updated

Indexes

Unique Index	1. uid 2. userid
--------------	---------------------

Users in groups table (ARSUSRGRP)

This table maintains the association of groups with individual users.

Table 36. Users in groups table (ARSUSRGRP)

Column Name	Data Type	Description
gid	INTEGER	Group identifier
uid	INTEGER	User identifier

Indexes

Unique Index (Oracle, SQL Server)	gid, uid
Unique Cluster Index (DB2)	gid, uid

User group ID table (ARSUSRGRPID)

This table maintains the association of individual users with user owners and their authority for groups.

Table 37. User group ID table (ARSUSRGRPID)

Column Name	Data Type	Description
id	INTEGER	User identifier
owner_id	INTEGER	Identifier of the owner of the user
perms	INTEGER	User authority for the group

Indexes

Unique Index (Oracle, SQL Server)	id, owner_id
Unique Cluster Index (DB2)	id, owner_id

Application group data table

An application group data table contains one row for each document that is stored in the application group. The name of an application group data table is `owner.applgrpInternalIdn`, where `owner` is the value of the `SRVR_INSTANCE_OWNER` parameter for the instance from the `ARS.INI` file, `applgrpInternalId` is the identifier that is assigned to the application group when the application group is created, and `n` is an incremental value. For example, `arsdbas3.rba1`, `arsdbas3.rba2`, and so on. The application group identifier is listed on the Advanced Storage Management page of the application group.

The application group data tables are also known as dynamic tables that are created by OnDemand.

A row in an application group data table contains from one to 32 user-defined fields followed by approximately 40 bytes of OnDemand control information (fields).

Table 38. Application group data table

Column Name	Data Type	Description
<code>user_field_1</code>	varies	First user-defined application group field
<code>user_field_n</code>	varies	Last user-defined application group field
<code>doc_name</code>	VARCHAR(11)	Document name
<code>doc_off</code>	INTEGER	Document offset
<code>doc_len</code>	INTEGER	Document length
<code>comp_off</code>	INTEGER	Compression offset
<code>comp_len</code>	INTEGER	Compression length
<code>annot</code>	CHAR(1) FOR BIT DATA	Annotation flag
<code>comp_type</code>	CHAR(1) FOR BIT DATA	Compression type
<code>resource</code>	INTEGER	Resource identifier
<code>pri_nid</code>	SMALLINT	Primary storage node identifier
<code>sec_nid</code>	SMALLINT	Secondary storage node identifier

Indexes This table is indexed on one or more of the user-defined fields:

Index	<code>user_field_1 ... user_field_n</code>
-------	--

Chapter 28. Table space file systems

When configuring DB2 table space file systems under Solaris, each file system that you define to contain SMS table spaces must be located on a different mount point. For example:

Table 39. SMS Table Space Files Systems under Solaris

Directory in ARS.DBFS	Mount Point
/arsdb/db1/SMS	/dev/HDISK6
/arsdb/db2/SMS	/dev/HDISK7

If you define more than one SMS table space file system on the same mount point, the OnDemand server will fail to initialize.

Chapter 29. TSM information

TSM Version 5.1

OnDemand Version 7.1.0.7 and later has been tested with the TSM Version 5.1 server, TSM Version 5.1 API, and TSM Version 5.1 Administrative Client on all platforms (AIX, HP-UX, Solaris, and Windows) and operates correctly.

Note: Customers who plan to run OnDemand under a 64-bit AIX operating system must use the 32-bit TSM API that is provided with TSM Version 4.2.2.1 or later. For TSM Version 4.2.2.1 or TSM Version 5.1.1, the 32-bit API can be found in the AIX43 section of the Client package. For TSM Version 5.1.5, there are separate packages for the 32-bit client and the 64-bit client. The TSM Client is available from IBM service on the Web at <ftp://service.boulder.ibm.com/storage/tivoli-storage-management/maintenance/client>.

See the TSM Version 5.1 *Quick Start* publication for your server operating system for information about migrating, installing, configuring, and using TSM.

Specifying license information in TSM Version 5.1 is the same as in TSM versions 4.1 and 4.2 (see “TSM Version 4.1” on page 126).

Beginning with TSM Version 5.1, the SCSI device special file names are now specified in the DEFINE PATH and UPDATE PATH commands, rather than in the DEFINE DRIVE, UPDATE DRIVE, DEFINE LIBRARY, and UPDATE LIBRARY commands, as shown in the examples of defining libraries and drives in the OnDemand publications. (The examples in the OnDemand publications are still valid for TSM versions 3.7, 4.1, and 4.2.)

1. For example, the OnDemand publications show the following command to define a library to TSM.

```
def library archlib0 libtype=scsi device=/dev/lb0
```

With version TSM Version 5.1, use the DEFINE PATH command to specify the device names of SCSI and 349X libraries and the external media manager paths for EXTERNAL libraries. For example:

```
def library archlib0 libtype=scsi
def path archive archlib0 srctype=server -
  desttype=library device=/dev/lb0
```

2. For example, the OnDemand publications show the following commands to define the drives in a library to TSM.

```
def drive archlib0 optdrv0 device=/dev/rop0 element=1
def drive archlib0 optdrv1 device=/dev/rop1 element=2
def drive archlib0 optdrv2 device=/dev/rop2 element=3
def drive archlib0 optdrv3 device=/dev/rop3 element=4
```

With version TSM Version 5.1, use the DEFINE PATH command to specify the device names of each of the drives in the library. For example:

```
def drive archlib0 optdrv0 element=1
def drive archlib0 optdrv1 element=2
def drive archlib0 optdrv2 element=3
def drive archlib0 optdrv3 element=4
```

```
def path archive optdrv0 srctype=server desttype=drive -
```

```
library=archlib0 device=/dev/rop0
def path archive optdrv1 srctype=server desttype=drive -
library=archlib0 device=/dev/rop1
def path archive optdrv2 srctype=server desttype=drive -
library=archlib0 device=/dev/rop2
def path archive optdrv3 srctype=server desttype=drive -
library=archlib0 device=/dev/rop3
```

Note: When upgrading to TSM Version 5.1, the installation program automatically updates existing LIBRARY and DRIVE definitions to include the PATH definitions.

3. The other TSM commands that are used by OnDemand, such as DEFINE DEVCLASS, REGISTER NODE, DEFINE COPYGROUP, and so on are unchanged in TSM Version 5.1.

See Adding a Manual Library and Adding an Automated Tape Library in the TSM Version 5.1 Administrator's Reference for more information and examples.

TSM Version 4.1

License registration has changed beginning with TSM Version 4.1.

- You no longer have to register a network license.
- The client and extended device support license names have changed from those documented in the *Installation and Configuration Guide* for your server operating system (GC27-0834-00 for UNIX or GC27-0835 for Windows). A client is now a Managed System; devices that required extended (or advanced) device support now require a Managed Library license.
- The following example shows how to register the licenses for ten managed systems (known as client nodes in TSM Version 3.7) and one managed library for an optical library that requires extended device support on a TSM Version 4.1 server:

```
register license file(10mgsyslan.lic)
register license file(1library.lic)
```

See the TSM Version 4.1 *Quick Start* publication for your server operating system for more information about registering licenses.

Additional information on device categories can be found at the Tivoli Storage Manager Web site: http://www.tivoli.com/support/storage_mgr/tivolimain.html. Follow the links to device information.

Deleting application groups

If the owner of the OnDemand instance is not root, when a user deletes an application group, OnDemand will delete the application group table(s) from the OnDemand database but will not delete the application group data from TSM. In this case, OnDemand issues message number ARS0022, which states that the user must manually delete the application group data from TSM. To delete the data from TSM, log on to TSM and use TSM utilities to delete the filespace in TSM that is associated with the application group. The name of the filespace is specified in message number ARS0022. See your TSM information for details about logging on to TSM and using TSM utilities to delete data.

Chapter 30. Web Enablement Kit

This section contains additions and corrections to the *IBM Content Manager OnDemand Version 7.1: Web Enablement Kit Implementation Guide*, SC27-1000-02.

Note: Beginning with Version 7.1.0.13, an updated version of *IBM Content Manager OnDemand Version 7.1: Web Enablement Kit Implementation Guide*, SC27-1000-02 is now available. The updated version contains all of the additions and corrections to the publication that were included in these release notes from 7.1.0.0 through 7.1.0.12. You can obtain a copy of the updated version of the *Web Enablement Kit Implementation Guide* directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

AFP Web Viewer

The following settings may be applied from logical views on the server to the AFP Web Viewer in Version 7.1.0.14.

- **Background Color.** The following colors are supported. No other colors are supported.
 - Green Bar (displayed with a white background)
 - Green
 - Red
 - Yellow
 - Black
 - White
 - Grey
- **Image Color.** The following colors are supported. No other colors are supported.
 - Yellow
 - Blue
 - Red
 - Magenta
 - Green
 - Cyan
 - Default (should display as black)
- **Zoom.**

Note: Selected Area Color is not applied to the AFP Web Viewer. The selected area is always displayed with white text and a black background.

ARSWWW.INI file

This section contains updates to the information about parameters that may be specified in the ARSWWW.INI file.

AddFieldsToDocId

The AddFieldsToDocId parameter is not supported when accessing an OnDemand/390 Version 2 server. ODWEK will automatically turn this option off when accessing an OnDemand/390 Version 2 server.

ShowSearchString

The ShowSearchString parameter is new in Version 7.1.0.13. The ShowSearchString parameter determines whether the Auto Find function is active or inactive. The Auto Find function supports transaction and text searching of line data documents from the Java line data viewer. The Auto Find function will automatically find and highlight the specific line in any document that matches the search criteria specified by the user.

When the Auto Find function is activated and a user performs either a transaction or text search and opens a document from the resulting document list, the system automatically searches the text of the document for the specified search criteria. If the search criteria is found, the line containing the search criteria is highlighted; otherwise the appropriate message is displayed. When the user opens another document for viewing (or reopens a previously viewed document), the search is performed again.

To activate the Auto Find function, set the ShowSearchString parameter to 1 (one). To deactivate the Auto Find function, set the ShowSearchString parameter to 0 (zero).

Example:

```
[Configuration]
ShowSearchString=1
```

Notes:

1. The ShowSearchString parameter has a global scope. You specify it only once in the CONFIGURATION section.
2. The ShowSearchString parameter is optional. If not specified, the default value is 0 (zero; inactive).

CGI API reference

Document Hit List

The following parameter is missing from Table 12 on page 91 of the *IBM Content Manager OnDemand Version 7.1: Web Enablement Kit Implementation Guide*, SC27-1000-02.

Name=Value	Purpose
<i>folder field name</i> OP= <i>value</i>	The operator to be used to override the default operator for a folder search field. The <i>value</i> must be one of the following: 1 to indicate Equal 2 to indicate Not Equal 4 to indicate Less Than 8 to indicate Less Than or Equal 16 to indicate Greater Than 32 to indicate Greater Than or Equal 64 to indicate In 128 to indicate Not In 256 to indicate Like 512 to indicate Not Like 1024 to indicate Between 2048 to indicate Not Between

Example:

`http://www.xyz.com/cgi-bin/arswww.cgi?_function=dochitlist ... &accountOP=256&account=%33 ...`

Configuring ODWEK for DBCS languages

The CODEPAGE and LANGUAGE parameters in the ARSWWW.INI file are used to specify National Language (NL) configuration options.

The CODEPAGE parameter identifies the code page of the ODWEK server and needs to be compatible with OnDemand database on the OnDemand library server. The CODEPAGE parameter need be specified **only** if the code page of the workstation on which you are running your ODWEK application is different than the code page of the OnDemand database on the OnDemand library server. The system uses the code page of the workstation on which the ODWEK application is running as the default value.

The LANGUAGE parameter determines the message catalog that ODWEK uses to display messages.

Table 40 lists the DBCS code pages and languages supported by OnDemand. The **CODEPAGE=** column lists the value for the code page, and need be specified **only** if the code page of the workstation on which you are running your ODWEK application is different than the code page of the OnDemand database. The **LANGUAGE=** column lists the values that are associated with the translated message catalogs.

Note: Linux is not listed in the table because Linux cannot be the target server for an ODWEK application. That is, although ODWEK may run on a Linux system, the OnDemand database (OnDemand library server) cannot run on a Linux system.

Table 40. DBCS code pages, languages, code sets and locales

Territory	LANGUAGE=	OS	Database Code Page	CODEPAGE=	Code Set	Locale
China (PRC)	CHS	AIX	1383	1383	IBM_eucCN	zh_CN
		HP-UX	1383	1383	hp15CN	zh_CN.hp15CN
		Solaris	1383	1383	gb2312	zh
		Windows	1386	1386	GBK	—
		Host (EBCDIC)	935	935	IBM-935	—
Japan	JPN	AIX	954	954	IBM_eucJP	ja_JP
		HP-UX	954	954	eucJP	ja_JP.eucJP
		Solaris	954	954	eucJP	ja
		Windows	943	943	IBM-943	—
		Host (EBCDIC)	939	939	IBM-939	—
Korea, South	KOR	AIX	970	970	IBM_eucKR	ko_KR
		HP-UX	970	970	eucKR	ko_KR.eucKR
		Solaris	970	970	5601	ko
		Windows	1363	1363	1363	—
		Host (EBCDIC)	933	933	IBM-933	—

Table 40. DBCS code pages, languages, code sets and locales (continued)

Territory	LANGUAGE=	OS	Database Code Page	CODEPAGE=	Code Set	Locale
Taiwan	CHT	AIX	964	964	IBM_eucTW	zh_TW
		HP-UX	964	964	eucTW	zh_TW.eucTW
		Solaris	964	964	cns11643	zh_TW
		Windows	950	950	big5	—
		Host (EBCDIC)	937	937	IBM-937	—

For more information about configuring the OnDemand system for DBCS languages, see “National Language Support” in the *Installation and Configuration Guide* for your operating system.

Image Web Viewer

The following change was made to the Image Web Viewer in Version 7.1.0.14.

For multi-page images, when the vertical scroll bar tab is dragged a small window will appear next to the tab. This window will display the page number corresponding to the tab position and the number of pages in the image. For example, a display of **5 / 10** indicates that there are ten pages in the image and that, if the tab is released, page number five will become the current page.

This behavior can be suppressed by a registry setting within this key:

HKEY_LOCAL_MACHINE\Software\IBM\OnDemand Image Web Viewer\Preferences

If the string value `PageNumberScroll` is set to 0 (zero), the page number window will not be displayed when the scroll bar tab is dragged.

Within the same registry key, if the string value `PageNumberToolbar` is set to 1 (one), page number information will be displayed on the toolbar for multi-page images. For example, a display of **3 / 5** indicates that there are five pages in the image and that page number three is the current page.

Internet Explorer

The following problem occurs when using Internet Explorer 6.0 to access an ODWEK application on the Internet (as opposed to a local intranet).

When retrieving AFP documents from the server and converting them to HTML pages for viewing, Internet Explorer 6.0 does not render the converted HTML pages. The reason is the enhanced security in Internet Explorer 6.0, which no longer permits a browser page running an applet or JavaScript to retrieve and display pages that do not come from the original page. In ODWEK, the applet comes from the mid-tier server and attempts to display a page from the local cache. Internet Explorer 6.0 considers this as a security violation, even if the applet issuing the request is signed.

Use the information in Table 41 on page 131 to determine if your system is affected and to resolve the problem.

Table 41. Internet Explorer 6.0 AFP to HTML Conversion - problem, determination, action

Problem	Internet Explorer 6.0 fails to render AFP to HTML converted pages.
Determination	You are affected by this problem if you are trying to use Internet Explorer 6.0 to view HTML pages that were generated by the AFP to HTML transform.
Action	<p>Add the site from where the applet originated to the list of trusted sites or intranet sites (which ever is relevant) in Internet Explorer 6.0. To add to the list of trusted sites:</p> <ol style="list-style-type: none"> 1. Start Internet Explorer 6.0. 2. From the Tools menu, click Internet Options. 3. Click the Security tab. 4. Select Trusted Sites in the list of zones. 5. Click Sites. 6. Type the relevant site URL in the Add this Web site to the zone field. If the site is not https://, then clear the Require server verification (https:) for all sites in this zone check box. 7. Click Add. 8. Click OK. 9. Close Internet Explorer. <p>Note: Adding to the list of intranet sites is similar, except after Step 4 above, there is an intermediate dialog box, with an Advanced button. Clicking the Advanced button will yield the same dialog box as in Step 5 above, and the remaining steps are the same.</p>

Java APIs

The following changes were made to the Java APIs in Version 7.1.0.14.

1. A new method, **isRequired()**, has been added to the ODCriteria object. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public boolean **isRequired()**

Returns: A boolean value indicating whether a value is required for this criteria when searching the folder.

2. A new method, **isDefaultValueAvailable()**, has been added to the ODCriteria object. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public boolean **isDefaultValueAvailable()**

Returns: A boolean value indicating whether a default value is available for this criteria.

3. A new method, **isDefaultValueFixed()**, has been added to the ODCriteria object. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public boolean **isDefaultValueFixed()**

Returns: A boolean value indicating whether the default value for this criteria is fixed (cannot be changed).

4. A new method, **getMaxEntryChars()**, has been added to the ODCriteria object. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public boolean **getMaxEntryChars()**

Returns: The maximum number of characters that can be used for this criteria during a search operation.

5. A new method, **getMaxDisplayChars()**, has been added to the ODCriteria object. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public boolean **getMaxDisplayChars()**

Returns: The maximum number of characters that will be present in a display value for this criteria after a search operation.

6. A new variation of the logon method, **logon(String, String, String, String)**, has been added to the ODServer object. This method can be used to update a password if the current password has expired. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public void **logon(String server,
String user,
String password,
String new_password)**

Parameters: server

The name of the OnDemand server.

user

The logon userid.

password

The current password for the specified userid.

new_password

A new password to be set if the current password has expired.

The following changes were made to the Java APIs in Version 7.1.0.13.

1. A new method, **getNoteStatus(boolean allow_unknown)**, has been added to the ODHit object. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public int **getNoteStatus(boolean allow_unknown)**

Parameters: allow_unknown

If true, and the server must be accessed to determine the status, Unknown will be returned. If false, the server will be accessed to determine the status if necessary.

Returns: An integer value indicating the status of notes for the document represented by the ODHit object.

2. A new variation of the search method, **search(ODCallback, boolean)**, has been added to the ODFolder object. This method must be used for searching whenever a parameter specifying the document location is to be returned to a HitCallback method. The ODApiDoc.zip file contains more detailed documentation.

Syntax: public Vector **search(ODCallback odCallback, boolean use_docloc_hit_callback)**

Parameters: odCallback

An instance of the `ODCallback` object.

use_docloc_hit_callback

Indicates whether (true) or not (false) to return a parameter specifying the document location to the HitCallback method.

Returns: A vector containing ODHit objects.

Shared library

Table 1 on page 28 and Table 20 on page 120 of the *IBM Content Manager OnDemand Version 7.1: Web Enablement Kit Implementation Guide, SC27-1000-02* incorrectly names several of the shared libraries. The correct names of the shared library files are listed in Table 42.

Table 42. Shared Library

Operating System	Shared Library
AIX	libarswwwsl.a
HP-UX	libarswwwsl.sl
Linux	libarswwwsl.so
Solaris	libarswwwsl.so
Windows	arswwwsl.dll

Chapter 31. Windows client customization

This section contains additions and corrections to the *Windows Client Customization Guide*, SC27-0837-01.

Note: Beginning with Version 7.1.0.8, an updated version of *IBM Content Manager OnDemand: Windows Client Customization Guide*, SC27-0837-01 is now available. The updated version contains all of the additions and corrections to the publication that were included in this release notes file from 7.1.0.0 through 7.1.0.7. You can obtain a copy of the updated version of the *Windows Client Customization Guide* directly from the Web at <http://www.ibm.com/software/data/ondemand/mp/library.html>.

The following methods have been added to the OLE Control:

- GetStoreDocInvalidFieldNum (see “GetStoreDocInvalidFieldNum” on page 136)
- SetResourceCacheMode (see “SetResourceCacheMode” on page 139)

The following methods have been updated:

- RetrieveDoc (see “RetrieveDoc” on page 144)
- StoreDoc (see “StoreDoc” on page 141)

The following DDE command has been updated:

- STORE_DOC (see “STORE_DOC” on page 144)

The behavior of the Windows client may be modified through various Registry entries. See “Modifying client behavior through the Registry” on page 144.

GetStoreDocInvalidFieldNum

short GetStoreDocInvalidFieldNum(
VARIANT * pFieldNum)

Parameters

pFieldNum

Points to a variable to receive the number of the invalid field. On return, this variable is set to type VT_I2.

Description

If a StoreDoc method was previously used and the return code was one of the following:

ARS_OLE_RC_INVALID_DATE_FIELD
ARS_OLE_RC_INVALID_INTEGER_FIELD
ARS_OLE_RC_INVALID_DECIMAL_FIELD
ARS_OLE_RC_TOO_MANY_VALUE_CHARS
ARS_OLE_RC_INVALID_APPLGRP_FIELD_TYPE,

the field number returned is the zero-based index of the folder field which caused the problem. If a previous StoreDoc method did not return one of the listed codes, the value received is unpredictable.

Return Value

Refer to "Return Code" in the *Windows Client Customization Guide* for an explanation of the return code.

See Also

StoreDoc

Examples

The following example retrieves the invalid field number following a StoreDoc.

C/C++ Example

```
CArsOle * pArsCtrl;
short rc;
SAFEARRAY * pSA;
VARIANT var;
BSTR bstrElement;
long i;
.
.
pSA = SafeArrayCreateVector(VT_BSTR, 0, 2);
if ( pSA == NULL )
    ERROR;

bstrElement = SysAllocStringByteLen ("255-546-667", 11);
i = 0;
SafeArrayPutElement (pSA, &i, bstrElement);
bstrElement = SysAllocStringByteLen ("06/07/94", 8);
i = 1;
SafeArrayPutElement (pSA, &i, bstrElement);

var.vt = VT_ARRAY | VT_BSTR;
var.parray = pSA;
rc = pArsCtrl->StoreDoc( "g:\\download \\file.afp",
                        "BKH-CRD",
                        "BKH-CRD",
                        &var );
if ( rc != ARS_OLE_RC_SUCCESS )
{
    if ( rc == ARS_OLE_RC_INVALID_DATE_FIELD |
        rc == ARS_OLE_RC_INVALID_INTEGER_FIELD |
        rc == ARS_OLE_RC_INVALID_DECIMAL_FIELD |
        rc == ARS_OLE_RC_TOOMANY_VALUE_CHARS |
        rc == ARS_OLE_RC_INVALID_APPLGRP_FIELD_TYPE )
    {
        VARIANT var1;
        rc = pArsCtrl->GetStoreDocInvalidFieldNum(&var1);
        .
        .
    }
    ERROR;
}
```

Visual Basic Example

```
Dim values(2) As String
Dim rc As Integer
.
.
.
values(0) = "255-546-667"
values(1) = "06/07/94"
var = values

rc = Ars01e.StoreDoc ("g:\download \file.afp", _
                    "BKH-CRD", _
                    "BKH-CRD", _
                    var)
if rc <> ARS_OLE_RC_SUCCESS Then
  if rc = ARS_OLE_RC_INVALID_DATE_FIELD or
  rc = ARS_OLE_RC_INVALID_INTEGER_FIELD or
  rc = ARS_OLE_RC_INVALID_DECIMAL_FIELD or
  rc = ARS_OLE_RC_TOOMANY_VALUE_CHARS or
  rc = ARS_OLE_RC_INVALID_APPLGRP_FIELD_TYPE then
    rc = Ars01e.GetStoreDocInvalidFieldNum (var1)
  End If
  MsgBox "ERROR"
  End
End If
.
.
.
```

SetResourceCacheMode

short **SetResourceCacheMode**(
short **Mode**)

Parameters

Mode

Specifies the new resource cache mode. This must be one of the following mode values found in ARSOLEEX.H:

ARS_OLE_RES_MODE_RETAIN

ARS_OLE_RES_MODE_ERASE_AFTER_RETRIEVE

ARS_OLE_RES_MODE_RETAIN specifies that any resource group retrieved during execution of a RetrieveDoc method be retained for use by subsequent RetrieveDoc or OpenDoc methods. Files containing these resource groups will not be erased until all instances of the control have been terminated. This is the default mode.

ARS_OLE_RES_MODE_ERASE_AFTER_RETRIEVE specifies that the file for any resource group retrieved during execution of a RetrieveDoc method be erased before returning to the caller. If subsequent RetrieveDoc or OpenDoc methods require access to the same resource group, another retrieval will be required.

Description

The requested mode is set.

Return Value

Refer to "Return Code" in the *Windows Client Customization Guide* for an explanation of the return code.

See Also

RetrieveDoc

Examples

The following example sets the resource cache mode to retain.

C/C++ Example

```
CArsOle * pArsCtrl;  
short rc;  
.  
.  
rc = pArsCtrl->SetResourceCacheMode( ARS_OLE_RES_MODE_RETAIN );  
if ( rc != ARS_OLE_RC_SUCCESS )  
    ERROR;  
.  
.
```

Visual Basic Example

```
Dim rc As Integer  
.  
.  
rc = ArsOle.SetResourceCacheMode (ARS_OLE_RES_MODE_RETAIN)  
if rc <> ARS_OLE_RC_SUCCESS Then
```

```
        MsgBox "ERROR"  
    End  
End If  
.  
.
```

StoreDoc

```
short StoreDoc(  
    char * DocPath,  
    char * ApplGrpName,  
    char * ApplName,  
    VARIANT * Values )
```

Parameters

DocPath

Specifies the fully-qualified path of a file containing the document data to be stored in the OnDemand database. This parameter is required.

ApplGrpName

Specifies the name of an Application Group within the folder. It is the responsibility of the caller to know the Application Group names associated with the active folder. This parameter is required.

ApplName

Specifies the name of an Application within the specified Application Group. It is the responsibility of the caller to know the Application names associated with the specified Application Group. This parameter is required.

Values

Points to a SafeArray of folder values. Each value is a character string which will be converted to data of the field type (that is, integer, date, and so on).

The number and order of folder fields can be determined by using the GetFolderFieldName(s) methods. For more information on these methods, see "GetFolderFieldName" and "GetFolderFieldNames" in the *Windows Client Customization Guide*.

Any folder fields not specified are given an empty string for string fields or zero for numeric fields. If extraneous fields are specified, they are ignored.

Date fields must be provided in the format required for the field (for example, specifying 02/03/96 is invalid when February 3, 1996 is required).

Description

OnDemand converts the folder field values to application group fields and stores the data from the specified file in the database as a document associated with the specified Application Group and Application.

If the return code is one of the following:

```
ARS_OLE_RC_INVALID_DATE_FIELD  
ARS_OLE_RC_INVALID_INTEGER_FIELD  
ARS_OLE_RC_INVALID_DECIMAL_FIELD  
ARS_OLE_RC_TOO_MANY_VALUE_CHARS  
ARS_OLE_RC_INVALID_APPLGRP_FIELD_TYPE,
```

the GetStoreDocInvalidFieldNum method can be used to determine the folder field which caused the problem.

Return Value

Refer to "Return Code" in the *Windows Client Customization Guide* for an explanation of the return code.

See Also

GetNumFolderFields, GetFolderFieldName, GetFolderFieldNames,
GetStoreDocInvalidFieldNum

Examples

C/C++ Example

```
CArsOle * pArsCtrl;  
short rc;  
SAFEARRAY * pSA;  
VARIANT var;  
BSTR bstrElement;  
long i;  
.  
.  
pSA = SafeArrayCreateVector(VT_BSTR,0,2);  
if ( pSA == NULL )  
    ERROR;  
  
bstrElement = SysAllocStringByteLen ("255-546-667",11);  
i = 0;  
SafeArrayPutElement (pSA, &i,bstrElement);  
bstrElement = SysAllocStringByteLen ("06/07/94",8);  
i = 1;  
SafeArrayPutElement (pSA, &i,bstrElement);  
  
var.vt = VT_ARRAY | VT_BSTR;  
var.parray = pSA;  
  
rc = pArsCtrl->StoreDoc("g:\\download \\file.afp",  
                        "BKH-CRD",  
                        "BKH-CRD",  
                        &var);  
if (rc != ARS_OLE_RC_SUCCESS)  
    ERROR;  
.  
.  
.
```


Visual Basic Example

```
Dim values(2) As String
Dim rc As Integer
.
.
.
values(0) = "255-546-667"
values(1) = "06/07/94"
var = values
rc = ArsOle.StoreDoc ("g:\download \file.afp", _
                    "BKH-CRD", _
                    "BKH-CRD", _
                    var)
if rc <> ARS_OLE_RC_SUCCESS Then
    MsgBox "ERROR"
End
End If
.
.
.
```

RetrieveDoc

Add the following to the **See Also** list:

SetResourceCacheMode

STORE_DOC

Add the following Return Code:

27 ARS_DDE_RC_TOO_MANY_VALUE_CHARS

Add the following Return Data:

ARS_DDE_RC_TOO_MANY_VALUE_CHARS

Modifying client behavior through the Registry

Note: The following procedure requires that you edit the registry on the computer. You should not edit the registry unless it is absolutely necessary. If there is an error in the registry, the computer may not function properly. Before you proceed, you should make a backup copy of the registry and you should be familiar with how to restore the registry to the same version you were using when you last successfully started the computer. For instructions, see your Windows information.

The following registry values are available to modify the behavior of the Windows client. The values must be placed in this registry key:

HKEY_CURRENT_USER\Software\IBM\OnDemand32\Client\Preferences

FORCE_DEF_CHARSET If set to 1, this value forces the use of the default character set when selecting the document list font.

Default value: 0

GREEN_BAR_COLOR May be used to override the color of the “green” bands of the Green Bar background color. It must be specified as an RGB value. For example, to use gray bars instead of green, specify 192,192,192.

Default value: 128,255,128

LINEDATA_FONT_FACE_1
LINEDATA_FONT_FACE_2
LINEDATA_FONT_FACE_3

May be used to override the initial font used for line data documents. The first time that a line data document is opened after OnDemand is initially installed, the client selects a font. If any of these values are specified, OnDemand will search first for the font face name specified by LINEDATA_FONT_FACE_1, then for LINEDATA_FONT_FACE_2, and finally LINEDATA_FONT_FACE_3 before trying other names.

For example, for LINEDATA_FONT_FACE_1, specify FixedSys

MAXIMIZE_WINDOW

If set to 1, maximizes the client window.

Default value: 0

MULTIFIND	<p>If set to 1, when performing a text find operation in an AFP document, all occurrences of the string on a page will be highlighted.</p> <p>Default value: 0</p>
SCS_CODEPAGE	<p>May be used to override the codepage used for SCS data by the AFP Viewer. If this value is 0, the codepage used is language dependent.</p> <p>Default value: 0</p>
SUPPRESS_FILE_CLEANUP	<p>If set to 1, prevents the deletion of obsolete files in data and resource directories during client initialization.</p> <p>Default value: 0</p>
SUPPRESS_WIN_POS_SAVE	<p>If set to 1, prevents the saving of window position and size during client termination.</p> <p>Default value: 0</p>
TRACE	<p>If set to 1, causes a trace file to be produced during AFP Viewer operations. The file is named VIEWER.LOG and is placed in the OnDemand project directory.</p> <p>Default value: 0</p>
TTONLY	<p>If set to 1, forces the AFP Viewer to use only True Type fonts.</p> <p>Default value: 0</p>
UNDEFINED_CP	<p>May be used to override the character displayed by the AFP Viewer in place of undefined code points. It must be specified as a hexadecimal character. For example, specify 2E for a period.</p> <p>Default value: 20</p>

Chapter 32. Windows server information

Windows Server 2003

Beginning with Version 7.1.0.15, OnDemand now supports Windows Server 2003.

OnDemand has been tested with Windows Server 2003 and the following products:

Product	Version	OnDemand Component
DB2 UDB	8.1.0	Server 7.1.0.14 or later
Oracle	9.2.0.1	Server 7.1.0.14 or later
SQL Server 2000	Service Pack 3	Server 7.1.0.15 or later
WebSphere Application Server	5.0.2	ODWEK 7.1.0.14 or later
IBM HTTP Server	1.3.26.2	ODWEK 7.1.0.14 or later
Windows	Windows Server 2003 Enterprise Edition	Server, clients 7.1.0.14 or later

ARSLOAD service failure

An OnDemand Load Data service (ARSLOAD) that was created by an OnDemand Version 7.1.0.6 or earlier Windows server may fail to start if the instance name is not ARCHIVE. Use the information in Table 43 to determine if your server is affected and to resolve the problem.

Table 43. ARSLOAD service failure - problem, determination, action

Problem	OnDemand Load Data Service (ARSLOAD) fails to start if the OnDemand instance is not named ARCHIVE.
Determination	Your server is affected by this problem if the OnDemand Load Data service (ARSLOAD) was created by an OnDemand Version 7.1.0.6 or earlier Windows server, the instance is not named ARCHIVE, and you have configured the service to automatically load data into that instance.

Table 43. ARSLOAD service failure - problem, determination, action (continued)

<p>Action</p>	<p>After installing the Version 7.1.0.7 or later PTF, you should take following actions to correct the problem. Use the OnDemand for Windows Configurator program to complete the actions. Note: The following steps need to be run only once after installing the 7.1.0.7 or later PTF and only if the instance name is not ARCHIVE.</p> <ol style="list-style-type: none"> 1. Start the OnDemand Configurator and select the server on which the affected ARSLOAD service is configured. 2. Stop the OnDemand Load Data (ARSLOAD) service. 3. Deinstall the OnDemand Load Data (ARSLOAD) service. 4. Reinstall the OnDemand Load Data (ARSLOAD) service. Note: IBM recommends that you reinstall the service by using the same userid that was defined to the services when you created the instance. 5. Restart the OnDemand Load Data (ARSLOAD) service. <p>See the online help in the OnDemand Configurator program for more information on these actions. If you have questions about this procedure or if you encounter any problems, please contact the IBM support center.</p>
----------------------	--

Database backup

The Configurator program that is provided with the OnDemand for Windows product allows for the scheduling of database backups. You can perform a backup while the database is either *online* or *offline*.

- If the backup is to be performed online, other applications or processes can continue to connect to the database, as well as read and modify data while the backup operation is running.
- If the backup is to be performed offline, only the backup operation can be connected to the database; other OnDemand services and the rest of your organization cannot connect to the database while the backup task is running.

To schedule an offline backup with the Configurator program, you must do the following:

1. Manually disconnect all other processes from the database before the backup task is scheduled to begin. This includes stopping the OnDemand LibSrvr, MVSD Server, and Load Data services on the library server. In addition, if you load data to the library server from another object server, then you should manually stop the OnDemand ObjSrvr and Load Data services on the object server.
2. Run the offline backup.
3. Verify that the offline backup completed successfully.
4. Manually restart the OnDemand LibSrvr service and the OnDemand MVSD Server and OnDemand Load Data services on the library server. If you stopped OnDemand services on an object server, manually restart the services.

Part 2. System administration

Chapter 33. Overview

Applications, application groups, folders, storage sets, and printers are the objects that represent how OnDemand stores, manages, retrieves, views, and prints reports and index data. Controlling and limiting access to the reports and index data is accomplished by defining users and groups and giving them the level of authority that is required to meet the data security strategy of an organization.

Chapter 34. User types

OnDemand provides the ability to centralize or decentralize the administration of the system. OnDemand also provides the flexibility to control access to objects from different levels. The most basic level of control is how the user is defined to the system. When a user is added, a user type is specified. Each user type has a different level of authority:

System Administrator

A system administrator has the highest level of authority on the system. A system administrator can perform all tasks on all of the objects that are defined to an OnDemand system. The objects are users, groups, applications, application groups, folders, storage sets, and printers. The tasks are add, update, delete, copy, export, create a report, and view properties. A system administrator also has the authority to modify the system parameters.

Application Group/Folder administrator

An application group/folder administrator has the authority to perform all tasks on all of the applications, application groups, and folders that are defined to an OnDemand system. The tasks are add, update, delete, copy, export, create a report, and view properties.

User Administrator

A user administrator has the authority to perform all tasks on all of the users that are defined to an OnDemand system. The tasks are add, update, delete, copy, export, create a report, and view properties. **Note:** A user administrator cannot create or delete a system administrator or an application group/folder administrator or change the user type of a system administrator or an application group/folder administrator.

User

A user has the lowest level of authority on the system. A user does not have access to any object on the system and therefore, cannot perform any tasks. The user must be given the authority to access an object and to perform a task on the object.

Chapter 35. Authority

A user's authority can be extended beyond the authority that is built into the user type of the user. Depending on the user type, up to four additional levels of authority can be given:

Create Users

An application group/folder administrator or a user can be given the authority to add users to the system. By default, when a user is added to the system, the user that performs the add task automatically has the authority to perform all other tasks on the user. The tasks are update, delete, generate a report, and view properties. When a user is added, the system automatically gives the user with Create Users authority the permission to access and administer the newly created user. (See Chapter 36, "Permissions," on page 159 for information about permissions at the user level.) If the permissions are later taken away, then the user with Create Users authority no longer has access to the user or the authority to administer the user.

A user with Create Users authority is similar to a user administrator in that both can create users. However, they differ because a user with Create Users authority can access and administer only the users that they create, so long as their access and administrator authority is not taken away. **Note:** A user with Create Users authority cannot create or delete a system administrator, an application group/folder administrator, or a user administrator or change the user type of a system administrator, an application group/folder administrator, or a user administrator.

A user with Create Users authority also has the authority to perform copy and export tasks on a user because both tasks add a user. An export task involves two userids: one on the server where the user exists and one on the server where the user will be added. The userid on the server where the user will be added must have the authority to add a user.

Create Groups

A user administrator, an application group/folder administrator, or a user can be given the authority to add groups to the system. The only other type of user that can add groups is a system administrator

By default, when a group is added to the system, the user with Create Groups authority is designated as the group owner. The group owner has the authority to perform all other tasks on the group. The tasks are update, delete, generate a report, and view properties. If the owner is changed to a different user or group, then the system automatically takes the authority to perform tasks on the group away from the user that originally created the group.

In general, a user has access to a group if the user is a system administrator, the owner of the group, a member of the group that has been designated as the owner, or a member of the group.

A user with Create Groups authority also has the authority to perform copy and export tasks on a group because both tasks add a group. An export task involves two userids: one on the server where the group exists

and one on the server where the group will be added. The userid on the server where the group will be added must have the authority to add a group.

Create Application Groups

A user administrator or a user can be given the authority to add application groups to the system. By default, when an application group is added to the system, the user that performs the add task automatically has the authority to perform all other tasks on the application group. The tasks are update, delete, generate a report, and view properties. When the application group is added, the system automatically gives the user with Create Application Groups authority the permission to access and administer the newly created application group. (See Chapter 36, "Permissions," on page 159 for information about permissions at the application group level.) If the permissions are later taken away, then the user with Create Application Groups authority no longer has access to the application group or the authority to administer the application group.

A user with Create Application Groups authority is similar to an application group/folder administrator in that both can create application groups. However, they differ because a user with Create Application Groups authority can access and administer only those application groups that they create, so long their access and administrator authority is not taken away. An application group/folder administrator can access and administer all of the application groups that are defined to the system.

Because applications are considered by OnDemand to be part of an application group, the permissions for accessing and administering applications are defined by the permission for the application group. For this reason too, a user with Create Application Groups authority can also create applications for the application group.

A user with Create Application Groups authority also has the authority to perform copy and export tasks on an application group because both tasks add an application group. An export operation involves two userids: one on the server where the application group exists and one on the server where the application group will be added. The userid on the server where the application group will be added must have the authority to add an application group.

Create Folders

A user administrator or a user can be given the authority to add folders to the system. By default, when a folder is added to the system, the user that performs the add task automatically has the authority to perform all of the other tasks on the folder. The tasks are update, delete, generate a report, and view properties. When a folder is added, the system automatically gives the user with Create Folders authority the permission to access and administer the newly created folder. (See Chapter 36, "Permissions," on page 159 for information about permissions at the folder level.) If the permissions are later taken away, then the user with Create Folders authority no longer has access to the folder or the authority to administer the folder.

A user with Create Folders authority is similar to an application group/folder administrator in that both can create folders. However, they differ because a user with Create Folders authority can access and administer only the folders that they create, so long as their access and

administrator authority is not taken away. An application group/folder administrator can access and administer all of the folders that are defined to the system.

A user with Create Folders authority also has the authority to perform copy and export tasks on a folder because both tasks add a folder. An export task involves two userids: one on the server where the folder exists and one on the server where the folder will be added. The userid on the server where the folder will be added must have the authority to add a folder.

Chapter 36. Permissions

Access and control of users, groups, applications, application groups, storage sets, folders, and printers can be given at various levels. In Chapter 34, "User types," on page 153, the level of control was determined by the user type of the user. In this section, a different level of control is described. Permissions on an object can be set from the Permissions page of the object. The following topics describe the objects and the permissions that can be set.

Users

A user can be given the authority to view documents that have been archived. A user can also be given the authority to perform administrative tasks on the system. A user can retrieve a list of users from the server with the administrative client. The list contains the users that the user has the authority to access. If the user is a system administrator or a user administrator, then the user also has the authority to administer any user in the list.

Access authority means the user can see the user in any list that contains users and can print or view the properties of the user. Access authority can be given to an individual user or to a group.

Access authority is especially helpful to application group/folder administrators, because they can give any user in the list access to application groups and folders. This is also true for group owners; access authority allows them to add users to the groups that they own.

Administrator authority of a user can be given to another user or to a group. Having administrator authority for a user means that the user or group given the authority can delete or update the user.

Groups

Groups can be created by a system administrator or a user that has Create Groups authority. Updating or deleting a group can be performed by a system administrator or the group owner. A group owner can be a user or another group. Allowing the group owner to be another group provides the ability to allow multiple users to administer the group.

To see a group in a list, a user must be a member of the group, the owner of the group, or a system administrator. For example, an application group/folder administrator can give groups access to application groups and folders. However, the application group/folder administrator must be able to see the groups in the list on the Permissions page of the application group or folder. This means that the application group/folder administrator must be a member of any group that requires access authority to an application group or a folder.

A system administrator, a group owner, or a member of the group can view the properties of the group and generate and print reports.

Applications

Because applications are considered by OnDemand to be part of an application group, the permissions for accessing and administering applications are defined by the permission for the application group. (See "Application Groups.")

Application Groups

Application Groups can be created by system administrators, application group/folder administrators, and users with Create Application Groups authority. After an application group is created, only a system administrator, an application group/folder administrator, a user with administrator authority for the application group, or a member of a group that has administrator authority for the application group can update or delete the application group. When a user with Create Application Groups authority creates an application group, the system automatically gives the user the authority to administer the application group. The user can update or delete the application group, so long as the administrator authority is not taken away.

To allow other users to see an application group in a list with the administrative client or to search for documents with the end-user client, the users must be given access authority to the application group. A user with access authority can also print or view the properties of the application group. Access is given on the Permissions page of an application group. There are three ways to give a user access to an application group:

1. Add the user's userid to the access list.
2. Add the name of a group to which the user belongs to the access list. The user and all of the other members of the group will have access to the application group.
3. Set the access permission for the reserved name *PUBLIC. All users on the system will have access to the application group. (The *PUBLIC name is used to set permissions for all users on the system.)

The levels of authority within the application group have a precedence order in which the permissions are enforced. The permissions that have been set for a user take precedence over any permissions that have been set for any groups that the user may belong to. User permissions also take precedence over permissions that have been set using the *PUBLIC name.

A user can also be given the authority to save a specific set of viewing attributes such as zoom, background color, and so forth. The viewing attributes can be used when a document is viewed with the end-user client. The set of viewing attributes, called a logical view, is accessible only to the user that created the logical view.

Document and Annotation permissions can also be set for users by using the *PUBLIC name, group names, and userids. Document permissions include add, delete, update, view, copy, print, and FAX. Annotation permissions include add, delete, update, view, and copy.

Folders

Folders can be created by system administrators, application group/folder administrators, and users with Create Folders authority. After a folder is created, only a system administrator, an application group/folder administrator, a user with administrator authority for the folder, or a member of a group that has

administrator authority for the folder can update or delete the folder. When a user with Create Folders authority creates a folder, the system automatically gives the user the authority to administer the folder. The user can update or delete the folder, so long as the administrator authority is not taken away.

To allow other users to see a folder in a list with the administrative client or to open a folder with the end-user client, the users must be given access authority to the folder. A user with access authority can also print or view the properties of the folder. Access is given on the Permissions page of a folder. There are three ways to give a user access to a folder:

1. Add the user's userid to the access list.
2. Add the name of a group to which the user belongs to the access list. The user and all of the other members of the group will have access to the folder.
3. Set the access permission for the reserved name *PUBLIC. All users on the system will have access to the folder. (The *PUBLIC name is used to set permissions for all users on the system.)

The levels of authority within the folder have a precedence order in which the permissions are enforced. The permissions that have been set for a user take precedence over any permissions that have been set for any groups that the user may belong to. The user permissions also take precedence over permissions that have been set using the *PUBLIC name.

In addition to allowing a user to access or administer a folder, a user can also be given the authority to customize the appearance of the folder search and display fields with the administrative client. The authority can be given to a specific user or to a group. If the authority is given to a group, then any member of the group has the authority to customize the appearance of the search and display fields. Only the authorized user or members of the group see the customized search and display fields with the end-user client. All other users will see the search and display fields that have been defined using the *PUBLIC name in the folder.

A user can also be given the authority to save a specific set of search criteria when using the end-user client. The user can restore the set of search criteria when needed, into the search fields of a folder. The set of search criteria, called a named query, can be made available to all of the users that have access to the folder (a public named query) or it can be made available only to the user that created the named query (a private named query). When giving Named Query authority to a user, the user can be given the authority to view named queries, but not create them; the user can also be given the authority to create public named queries, private named queries, or both. A user can get Named Query authority from a group, if the group has been given Named Query authority for the folder.

Storage sets

Any user on the system can view the properties of a storage set or generate a report about a storage set. However, only a system administrator can add, delete, update, copy, or export a storage set.

Printers

Printers are maintained by system administrators. Only a system administrator can add, delete, update, copy, or export a printer. By default, only system administrators can see printers in a list and therefore, only a system administrator can view the properties of a printer or generate a report about a printer. However,

it is almost always necessary for other users to have access to printers. For this reason, a user or group can be given access to a particular printer. A printer's access list is maintained on the Permissions page under printers, by using the administrative client. Access to a printer can be given to all users and groups defined to the system, individual groups, and individual users.

Limiting access to printers provides the ability to control which printers can be used to print archived documents by OnDemand users. For example, suppose that there is a printer in the customer service department. Only people in the customer service department should be permitted to print on the printer. You can accomplish this by creating an OnDemand group that contains only the department members and giving only that group access to the printer.

Chapter 37. System administration

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

The decision to centralize or decentralize the administration of the system should be made before objects are added to the system. While the decision is reversible, the amount of work required to change from one type of administration to the other can be significant if a large number of users, groups, folders, and application groups have already been added.

There are many ways to decentralize the administration of the system, because of the various user types and the additional authority levels that can be specified for users. Two specific models will be discussed in this section: the Object Type model and the Object Owner model.

- In the Object Type model, all of the objects on the system are logically grouped into administrative domains according to the type of the object. The administrator of a domain maintains all of the objects within the domain. For example, an application group/folder administrator maintains all of the application, application group, and folder objects on the system.
- In the Object Owner model, the objects on the system are logically grouped into administrative domains according to the creator/owner of the object. An administrator maintains only the objects that they create. For example, a user with create application groups and create folders authority can maintain only the applications, application groups, and folders that they created. The Object Owner model can be used to separate the objects on the system into logical parts, such as a department, a company, or some other entity. Each part is independent of the other and should be maintained separately. Each part typically requires two administrative users. One user has the responsibility for creating and maintaining users and groups. The other user has the responsibility for creating and maintaining applications, application groups, and folders. However, you can also define one user with the authority to create and maintain users, groups, applications, application groups, and folders. In effect, the one user would be the system administrator for a logical part of the system.

Object Type model

In the Object Type model, the system administrator defines two new users. One user is responsible for administering applications, application groups, and folders and is defined as an application group/folder administrator. The second user is responsible for administering users and groups and is defined as a user administrator with Create Groups authority. Table 44 on page 164 shows the administrative users and the tasks assigned to the users.

Table 44. Administrator Roles in the Object Type Model

User Type	Tasks
System Administrator	<ul style="list-style-type: none"> Create an application group/folder administrator Create a user administrator with Create Groups authority Create and maintain storage sets Create and maintain system printers
User Administrator with Create Groups authority	<ul style="list-style-type: none"> Create and maintain users Create and maintain groups
Application Group/Folder Administrator	<ul style="list-style-type: none"> Create and maintain application groups Create and maintain applications Create and maintain folders

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

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

To give an application group/folder administrator access to a user, the user administrator with Create Groups authority must update each user and give the application group/folder administrator access to the user. Once access has been given, the application group/folder administrator will be able to see the user in the list and will therefore be able to grant the user access to any application groups and folders on the system. Again, this is not the recommended approach because this task will have to be repeated each time that a user is added to the system.

Object Owner model

In the Object Owner model, the system administrator defines two users for each logical part of the system. One user is responsible for maintaining the users and groups for a logical part of the system. The other user is responsible for maintaining the applications, application groups, and folders for a logical part of the system. The Object Owner model allows you to store data from several sources

on one OnDemand system and let only one set of users access each set of data. Table 45 shows the administrative users and the tasks assigned to the users.

Table 45. Administrator Roles in the Object Owner Model

User Type	Tasks
System Administrator	<ul style="list-style-type: none"> Create a user with Create Users and Create Groups authority Create a user with Create Application Groups and Create Folders authority Create and maintain storage sets Create and maintain system printers
User with Create Users and Create Groups authority	<ul style="list-style-type: none"> Create and maintain users Create and maintain groups
User with Create Application Groups and Create Folders authority	<ul style="list-style-type: none"> Create and maintain application groups Create and maintain applications Create and maintain folders

In addition to the tasks listed in Table 45 under System Administrator, it is also necessary for the system administrator to give the user with Create Users and Groups authority access to the user with Create Application Groups and Folders authority. Otherwise, the user that creates groups will not be able to add the user that creates application groups and folders to any groups. To simplify the explanation that follows, the user with Create Users and Groups authority will be called the user administrator and the user with Create Application Groups and Folders authority will be called the application group/folder administrator.

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

Another reason for providing access to application groups and folders from a group rather than to an individual user is that the application group/folder administrator does not have access to the users and groups in this model. Since the application group/folder administrator must first be given access to any users or groups that require access to application groups or folders, it is simpler and less time consuming to give access to a few groups rather than hundreds or even thousands of users. The application group/folder administrator is given access to a group by adding the application group/folder administrator to the group. This is done by the user administrator in this model. As a group member, the application group/folder administrator will be able to see the group in the list and will therefore be able to grant the group access to the application groups and folders that have been defined by the application group/folder administrator in this model.

To give an application group/folder administrator access to a user, the user administrator must update each user and give the application group/folder administrator access to the user. Once access has been given, the application group/folder administrator will be able to see the user in the list and will therefore

be able to grant the user access to the application groups and folders. Again, this is not the recommended approach because this task will have to be repeated each time that a user is added to the system.

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

Chapter 38. Summary

There can be many different variations of the two models that have been described. For example, in the Object Owner model, rather than one user administering both application groups and folders, one user can be defined to administer application groups and another user can be defined to administer folders. Choosing the right model or variation is an important decision that should be made early in the planning process. Changing to a different model later is not impossible but may require additional work if there are a large number of objects defined on the system.

Chapter 39. Helpful hints

1. To simplify the task of providing access to application groups and folders, give access to a group rather than a user. When a new user needs access, add the user to the group.
2. To allow an application group/folder administrator to see groups in the permissions list, add the application group/folder administrator to the groups that require access to application groups and folders.
3. To allow multiple users to administer the same groups, create a group of users and make that group the group owner for any groups that need to be administered by multiple users.
4. The Create Groups authority is most effectively used if it is combined with the Create Users authority or added to a user administrator. Because the purpose of a group is to give a set of users permissions to another object, it is not very useful if the user that creates the group does not have access to any users. Otherwise, the user that creates a group must be given access to each user that needs to be added to the group.

Chapter 40. Local server setup

OnDemand supports two types of servers. The first type of server uses TCP/IP to communicate between the client programs and the server programs. The server programs run on operating systems, such as AIX, HP-UX, iSeries and OS/400, Sun Solaris, Windows², and z/OS and OS/390. The databases supported by the server are DB2, Oracle (AIX, HP-UX, Sun Solaris, Windows platforms only), and Microsoft SQL server (Windows platform only). Definitions of users, groups, applications, for example, are stored in the database as well as the index values for the report data that is loaded into the system.

The second type of server is a *local server*. The local server is self-contained (no TCP/IP communication) and is defined using files contained in a directory located on a PC rather than in a database. The files represent the system tables that define the various objects such as users, groups, and applications.

One of the uses of a local server is to provide the ability to export definitions from a non-local server to a local server so the definitions can then be imported to a different non-local server. This is especially beneficial when a TCP/IP connection does not exist between two non-local servers. One of the limitations of exporting definitions from a non-local server to a local server is the local server does not support any of the operating-specific or database-specific parameters that are defined when using the administrative client.

When a user logs on to a server, the administrative client determines whether the user has logged on to a local server or a non-local server. If the server is a non-local server, the operating system and the database types are determined. Based on the type of server, the operating system and the database are determined, default settings are established, entry fields are hidden or displayed, and values are added to or removed from selection lists. For definitions that have been exported from a non-local server to a local server, updating or viewing the definitions may not have the desired results because of the operating differences between local and non-local servers. For example, when an application group and application are exported from an OS/400 server to a local server, the OS/400 indexer that is specified in the application is not supported on a local server. When the application is viewed on the local server, the Indexer field on the Indexer Information page will not have a value and the OS/400 Indexer will not be listed as a choice of indexers.

To correctly display the operating system- and database-specific parameters on a local server from parameters that have been exported from a non-local server, two string values can be added to the local server definition in the Registry. The string values identify the operating system and the database. Then, when a user logs on to the local server from the administrative client, the server definitions will resemble and operate like a non-local server for the specified operating system and database. In the case of the application group and application that was exported from the OS/400 server, the OS/400 Indexer will now be selected when the application is updated or viewed from the local server. Because the OS/400

2. The OnDemand administrative client can run under Windows NT 4.0 with SP5 or later, Windows 2000, Windows XP and Windows 98.

Indexer is dependent upon the operating system and not the database type, only the operating system string value is necessary in the Registry.

As another example, an AIX server uses Oracle as the database. An application group and application are exported from the AIX server to a local server. To correctly display the database-specific parameters in the application group, the database string value of Oracle is added to the local server definition in the Registry.

Important: Using the Registry Editor incorrectly can cause serious problems that may require you to reinstall your operating system. IBM cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use the Registry Editor at your own risk.

For information about how to edit the registry, view the Changing Keys and Values Help topic in Registry Editor (Regedit.exe) or the Add and Delete Information in the Registry and Edit Registry Data Help topics in Regedt32.exe.

You should back up the registry before you proceed. If you are running Windows NT or Windows 2000, you should also update your Emergency Repair Disk (ERD).

To add an operating system value and a corresponding database value to the Registry, move to the
\\HKEY_CURRENT_USER\\Software\\IBM\\OnDemand32\\Common key. Under the Common key, move to the local server subkey that you want to modify. All server subkeys begin with the string @SRV@_. To specify an operating system for the local server, add a new value under the selected local server subkey. The name of the new value is OS_TYPE and the data type is String (REG_SZ). To specify a database for the local server, add a new value under the selected local server subkey. The name of the value is DB_TYPE and the data type is String (REG_SZ). The actual string values for the operating system and database are case insensitive and can be entered in lower, upper, or mixed case.

Table 46 contains a list of the string values that can be specified for the operating system value (OS_TYPE) and the corresponding database value (DB_TYPE). If a database is specified and the operating system does not support the database, the database value will be ignored and DB2 will be used. For example, if the OS_TYPE is OS400 and Oracle is specified for the DB_TYPE, the specified DB_TYPE (Oracle) will be ignored and DB2 will be used instead. If a database value is not specified, the default for the local server is DB2. If an operating system value is not specified, the default for the local server is AIX.

Table 46. Operating System and Database values

OS_TYPE	DB_TYPE
AIX	DB2, Oracle
HPUX	DB2, Oracle
OS390	DB2
OS400	DB2
SUN	DB2, Oracle
WIN32	DB2, Oracle, MSSQL

Figure 7 shows an example of the Registry after an operating system value and a database value have been added to a local server that is identified by the @SVR@_localserver subkey.

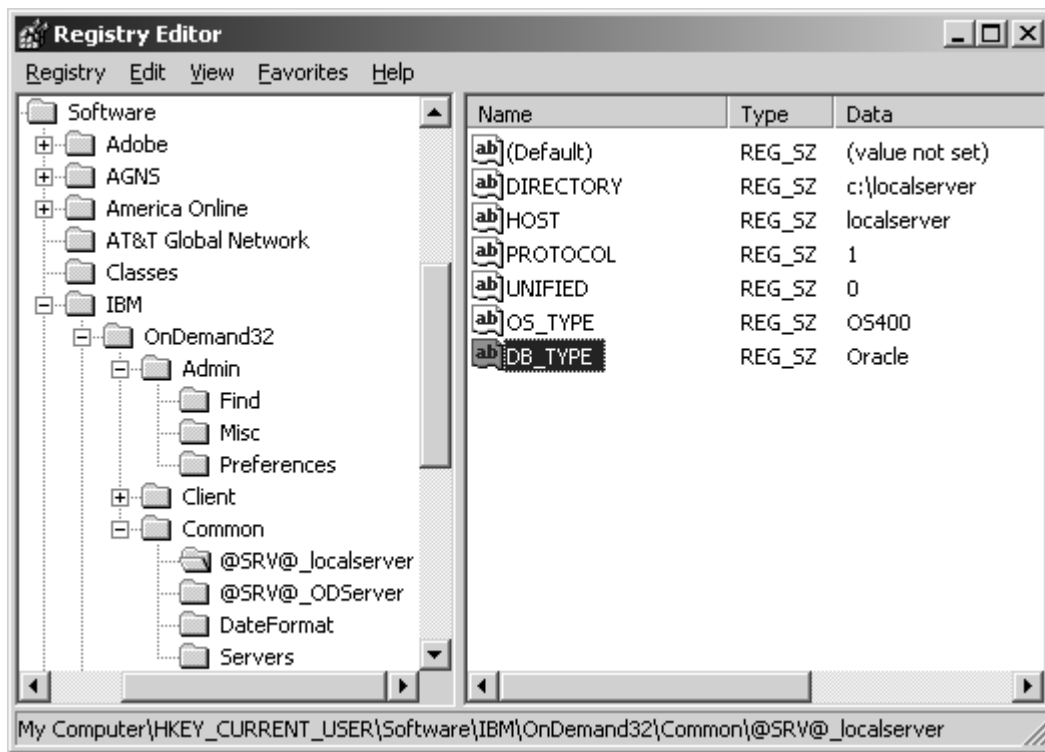


Figure 7.

Note: Although a local server can resemble a non-local server, the following are system limitations for a local server:

- The userid of a user that is defined on a local server cannot be updated.
- User and group permissions for users are not supported from the User Permissions page on Users dialog box.
- The Find function is not supported.
- Server Printers are not supported.
- Data Distribution Files and Groups are not supported.

Note: For a local server, the default administrative userid is admin. There is no initial password set for the admin userid on a local server.

Part 3. Xenos transforms

Note: In this publication, the term Xenos transform refers to the Xenos d2e Platform. Xenos d2e Platform is a trademark of Xenos Group Inc.

This part is organized into the following sections:

- General information. See Chapter 41, "Understanding Xenos," on page 177, Chapter 42, "Xenos transforms," on page 181, Chapter 43, "Loading data," on page 189, and Chapter 44, "Scenarios for using Xenos," on page 191.
- Information about indexing, transforming, and loading input files into the system. See Chapter 45, "How to specify parameters to the Xenos transforms," on page 197 and Chapter 46, "JS program reference," on page 203.
- Information about retrieving and transforming documents by using the OnDemand Web Enablement Kit (ODWEK). See Chapter 47, "Configuring ODWEK," on page 221.

Chapter 41. Understanding Xenos

Beginning with OnDemand Version 7.1, you can process input files and documents with transforms that are provided by Xenos. This section provides an overview of the Xenos transforms, explains the functions that the Xenos transforms can perform, and describes different scenarios for processing your input files and documents.

Important: Before you attempt to use the Xenos transforms on your system, you must obtain the transform program, license, and documentation from your IBM representative. Your IBM representative can also provide education and other types of help and support for processing data with the transform programs.

The Xenos transforms are batch application programs that let you process several different types of input print files or documents that are stored in the system. The Xenos transforms provide converting, indexing, and resource collecting capabilities that let you archive, retrieve, and view documents.

The Xenos transforms can be used in an OnDemand system:

- When loading input files into the system.
- When retrieving documents from the system by using ODWEK.

When loading input files into the system, you can use the Xenos transforms to:

- Convert AFP input files to PDF.
- Convert Metacode input files to AFP, Metacode, or PDF.
- Convert PCL input files to PDF.
- Index input files to enhance your ability to view, archive, or retrieve individual pages or groups of pages from large print files.
- For Metacode output, collect the resources needed for printing or viewing a document, so that you can print and view the exact document, possibly years after its creation.

The Xenos transforms process the input print data and resources and produce these files:

- Index file
- Document file
- Resource file (Metacode output only)

With the files that the Xenos transforms create, you can store the data into OnDemand and then use the Windows client to search for and retrieve, view, and print the documents. You can also use ODWEK to retrieve and transform documents.

When retrieving documents from the system by using ODWEK, you can use the Xenos transforms to:

- Convert AFP documents to HTML, PDF or XML files.
- Convert Metacode documents to AFP, HTML, PDF, or XML files.

The files that the Xenos transforms creates can then be sent to a Web browser for viewing and printing.

Figure 8 on page 179 shows a high-level overview of how the Xenos transforms fit into an OnDemand system for creating, indexing, viewing, and printing documents. The figure shows the resources and the AFP, Metacode, or PCL print data, which can be provided by various products, that can flow into the ARSLOAD program for processing. If the Indexer that is specified in OnDemand is Xenos, then the ARSLOAD program calls the Xenos transform with parameter and script files that you create. The files that the Xenos transform produces can then be processed by the ARSLOAD program for archiving and the client programs for viewing and printing. If you plan to use ODWEK, then you can transform AFP and Metacode documents that are stored in the system before sending them to the Web browser. For example, you could use the Xenos transform with the ARSLOAD program to process and load Metacode print files. Then, you could use ODWEK to retrieve a Metacode document from the system, call the Xenos transform to convert the Metacode document into a PDF file (by using other parameter and script files that you create), and send the PDF file to the browser.

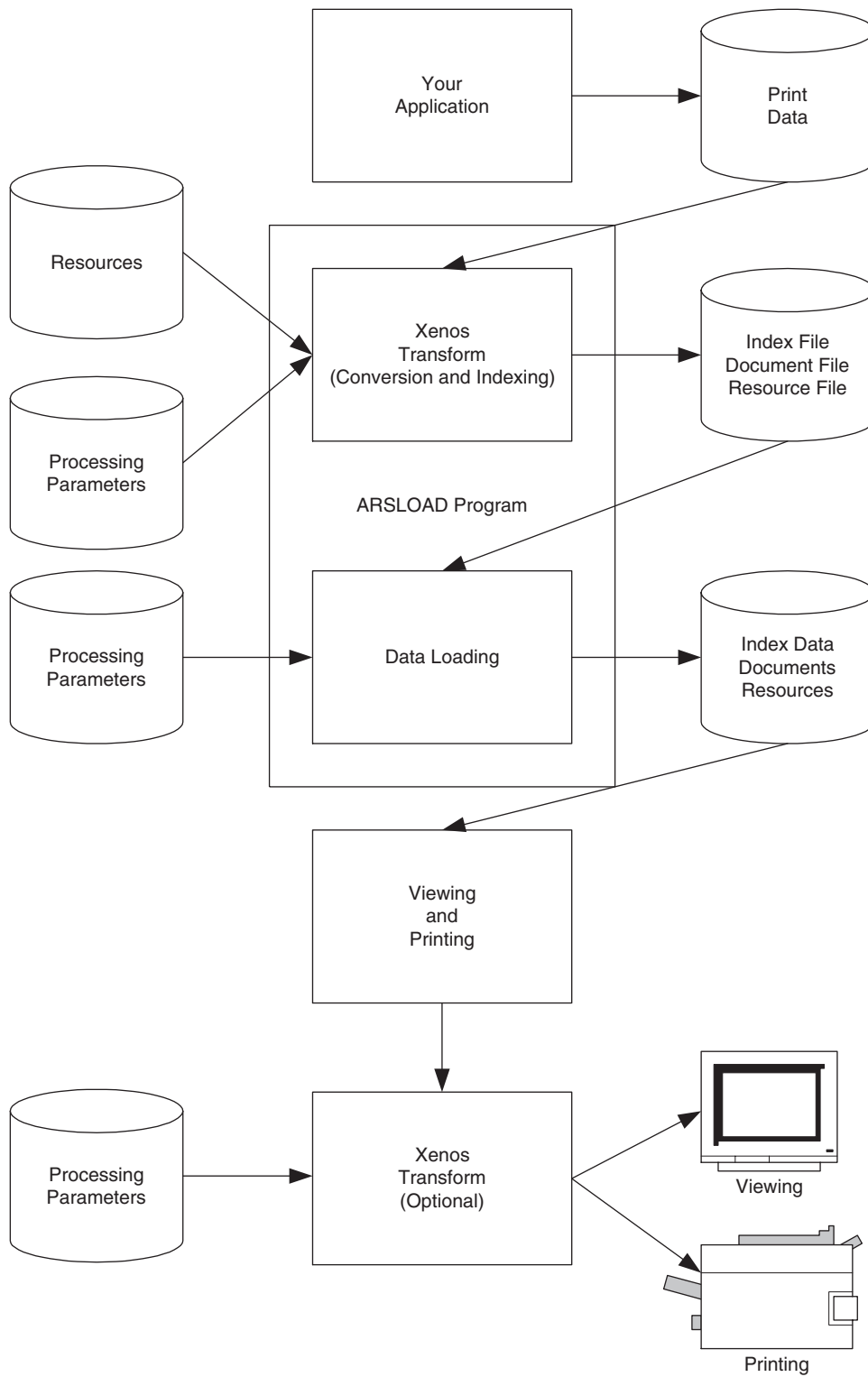


Figure 8. How the Xenos Transform fits into the OnDemand Environment

Chapter 42. Xenos transforms

You can use the Xenos transforms to perform these functions:

- Convert data streams
- Index documents
- Collect resources

Convert data streams

OnDemand customers can use the Xenos transforms for the following types of data conversions:

- Convert AFP documents to HTML
- Convert AFP data to PDF
- Convert AFP documents to XML
- Convert Metacode data to AFP
- Convert Metacode documents to HTML
- Convert Metacode data to PDF
- Convert Metacode input files to Metacode documents
- Convert Metacode documents to XML
- Convert PCL input files to PDF documents

The following sections describe each type of data. Please see the Xenos documentation to better understand the data formats and for more information about the transforms.

AFP documents to HTML

The AFP to HTML transform is used when retrieving AFP documents from an OnDemand server by using ODWEK.

There are two AFP to HTML transforms:

- The AFP to HTML/CSS transform converts AFP documents into HTML files with linked Cascading Style Sheets (CSS) for display with a Web browser. The output uses fonts from the user's system to display the pages, so it may not match the original as closely as other formats. Graphics are displayed using linked PNG image files. A separate HTML file is generated for each page in the original document. An optional HTML frameset file can also be generated to permit easier navigation between pages. The frameset is an HTML page that displays the individual HTML pages in a large frame at the top, and has links to the other pages in a small frame at the bottom.
- The AFP to HTML/Template Merger allows data from the input document to be inserted in a predefined template file. This allows the pages to be viewed to have a different format and layout than the printed page for redesigning the page to look and fit better on a standard computer screen. Multiple templates can be used in a single application, as needed, to suit the variety of information that is found from page to page in the input document.

AFP data to PDF

The AFP to PDF transform can be used when loading AFP input files into the system or retrieving AFP documents from the system by using ODWEK.

When loading AFP input files into the system, the transform can process linedata and mixed mode files that print on AFP printers through Infoprint or PSF as well as fully-composed MO:DCA-P files. The AFP to PDF transform converts AFP data streams into Adobe PDF documents. The PDF output can be viewed at the Windows client by using Adobe Acrobat or Adobe Acrobat Approval. To run the AFP to PDF transform when loading AFP input files into the system, set the Data Type on the View Information page to PDF and set the Indexer on the Indexing Information page to Xenos. Specify processing parameters on the Indexing Information page.

When retrieving AFP documents from the system by using ODWEK, the transform processes fully-composed MO:DCA-P files. The PDF output can be viewed at the Web browser by using an Adobe PDF viewer. To run the AFP to PDF transform when retrieving AFP documents from the system, set the AFPVIEWING parameter to XENOS in the ARSWWW.INI file and specify PDF as the OUTPUTTYPE in the ARSXENOS.INI file.

AFP documents to XML

The AFP to HTML transform is used when retrieving AFP documents from an OnDemand server by using ODWEK.

The AFP data to XML transform uses the AFP to HTML/Template Merger to allow data from the input document to be inserted in a predefined template file. This allows the pages to be viewed to have a different format and layout than the printed page, such as separating summary information from transaction details. The template is written in XML, for applications such as an online payment system. Multiple templates can be used in a single application, as needed, to suit the variety of information that is found from page to page in the input document.

Metacode to AFP

The Metacode to AFP transform can be used when loading Metacode input files into the system or retrieving Metacode documents from the system by using ODWEK.

When loading Metacode input files into the system, the transform converts Xerox Metacode and Linedata Conditioned Data Streams (LCDS) into AFP documents. Linedata and mixed mode files that print on Metacode Printers are supported, as well as pure Metacode files. The AFP output can be viewed at the Windows client. To run the Metacode to AFP transform when loading Metacode input files into the system, set the Data Type on the View Information page to AFP and set the Indexer on the Indexing Information page to Xenos. Specify processing parameters on the Indexing Information page.

When retrieving Metacode documents from the system by using ODWEK, the transform processes pure Metacode documents. The AFP output can be viewed at the Web browser by using the AFP Web Viewer. To run the Metacode to AFP transform when retrieving Metacode documents from the system by using ODWEK, set the METAVIEWING parameter to XENOS in the ARSWWW.INI file and specify AFP as the OUTPUTTYPE in the ARSXENOS.INI file.

Metacode documents to HTML

The Metacode to HTML transform is used when retrieving Metacode documents from an OnDemand server by using ODWEK.

There are two Metacode to HTML transforms:

- The Metacode to HTML/CSS transform converts Metacode documents into HTML files with linked Cascading Style Sheets (CSS) for display with a Web browser. The output uses fonts from the user's system to display the pages, so it may not match the original as closely as other formats. Graphics are displayed using linked PNG image files. A separate HTML file is generated for each page in the original document. An optional HTML frameset file can also be generated to permit easier navigation between pages. The frameset is an HTML page that displays the individual HTML pages in a large frame at the top, and has links to the other pages in a small frame at the bottom.
- The Metacode to HTML/Template Merger transform allows data from the input document to be inserted in a predefined template file. This allows the pages to be viewed to have a different format and layout than the printed page for redesigning the page to look and fit better on a standard computer screen. Multiple templates can be used in a single application, as needed, to suit the variety of information that is found from page to page in the input document.

Metacode to Metacode

The Metacode to Metacode transform is used when loading Metacode input files into the system.

The Metacode to Metacode transform converts Xerox Metacode and Linedata Conditioned Data Streams (LCDS) to Metacode documents. Linedata and mixed mode files that print on Metacode Printers are supported, as well as pure Metacode files. At first, the Metacode to Metacode transform might seem a bit redundant, converting a format that already prints on the destination printer to the same format. But the input to the transform can be inefficient linedata or very obscure metacode, where the resulting output Metacode is efficient and in a predictable format, which allows individual pages or groups of pages in the output Metacode to be indexed and retrieved.

To run the Metacode to Metacode transform when loading Metacode input files into the system, set the Data Type on the View Information page to Metacode and set the Indexer on the Indexing Information page to Xenos. Specify processing parameters on the Indexing Information page.

Metacode to PDF

The Metacode to PDF transform can be used when loading Metacode input files into the system or retrieving Metacode documents from the system by using ODWEK.

When loading Metacode input files into the system, the transform converts Xerox Metacode and Linedata Conditioned Data Streams (LCDS) into PDF documents. Linedata and mixed mode files that print on Metacode Printers are supported, as well as pure Metacode files. The PDF output can be viewed at the Windows client by using Adobe Acrobat or Adobe Acrobat Approval. To run the Metacode to PDF transform when loading Metacode input files into the system, set the Data Type on the View Information page to PDF and set the Indexer on the Indexing Information page to Xenos. Specify processing parameters on the Indexing Information page.

When retrieving Metacode documents from the system by using ODWEK, the transform processes pure Metacode documents. The PDF output can be viewed at the Web browser by using an Adobe PDF viewer. To run the Metacode to PDF transform when retrieving Metacode documents from the system by using ODWEK, set the METAVIEWING parameter to XENOS in the ARSWWW.INI file and specify PDF as the OUTPUTTYPE in the ARSXENOS.INI file.

Metacode documents to XML

The Metacode to XML transform is used when retrieving Metacode documents from an OnDemand server by using ODWEK.

The Metacode to XML transform uses the Metacode to HTML/Template Merger to allow data from the input document to be inserted in a predefined template file. This allows the pages to be viewed to have a different format and layout than the printed page, such as separating summary information from transaction details. The template is written in XML, for applications such as an online payment system. Multiple templates can be used in a single application, as needed, to suit the variety of information that is found from page to page in the input document.

PCL to PDF

The PCL to PDF transform is used when loading PCL input files into the system.

The PCL to PDF transform converts Hewlett Packard Printer Control Language (PCL) print files into Adobe PDF documents. The term PCL refers to the compound data stream used by the Hewlett Packard (HP) printers. The transform accepts most PCL 4 or 5 designed for HP desktop printers; the transform does not support the HP PGL or HP Deskjet formats. The PDF output can be viewed at the Windows client by using Adobe Acrobat or Adobe Acrobat Approval. The PDF output can also be viewed at a Web browser by using an Adobe PDF viewer.

To run the PCL to PDF transform when loading PCL input files into the system, set the Data Type on the View Information page to PDF and set the Indexer on the Indexing Information page to Xenos. Specify processing parameters on the Indexing Information page.

Index documents

The Xenos transforms can index input files. When indexing with the Xenos transforms, you can divide a large print file into smaller, uniquely identifiable units, called *groups*. For example, you can use the Xenos transforms to divide a large print file that was created by a bank statement application into individual groups by generating group indexes that define the group boundaries in the file. A group is a named collection of sequential pages, which, in this example, consists of the pages describing a single customer's account. For example, a bank statement application probably produces a large printout consisting of thousands of individual customer statements. You can think of these statements as smaller, separate units, each uniquely identifying an account number, date, Social Security number, or other attributes.

Using the Xenos transforms, you can create an OnDemand generic index file. The index file lets you:

- Retrieve individual statements from storage volumes, based on an account number or any other attribute.

- More rapidly access the statements for viewing by, for example, the Windows client.
- Archive individual statements or the entire indexed print file for long-term storage and subsequent data management and reprinting, even years after its creation.

The Xenos transforms can create a generic index file for the following types of input files:

- AFP data
- Metacode
- PCL data

The Xenos transforms let you generate the group indexes by extracting values that are present in the input data itself, when the data has been formatted so that the Xenos transforms can reliably locate the values. This kind of indexing is called *indexing with data values*.

Indexing with data values

Some applications such as payroll or accounting statements contain data that might be appropriate to use for indexing tags. In the bank statement example, the account number is a type of data value that you might want to tag. You can then store a single customer's account statement using the account number, and you can retrieve and view the same statement using the account number. If the data value that you want to use in an indexing tag is consistently located in the same place for each statement, then you can specify parameters to the Xenos transform that create a separate group of pages for each statement.

Collect resources

Resources can be collected only when running the Metacode to Metacode transform.

The Xenos transform can determine the list of resources needed to view the print file and collect the resources from the specified libraries. After you load the document and the resources into the system, you can view the document with fidelity.

Summary

Figure 9 shows the first part of the load process – you run the ARSLOAD program to process an input file and the Indexer that is specified to OnDemand is Xenos.

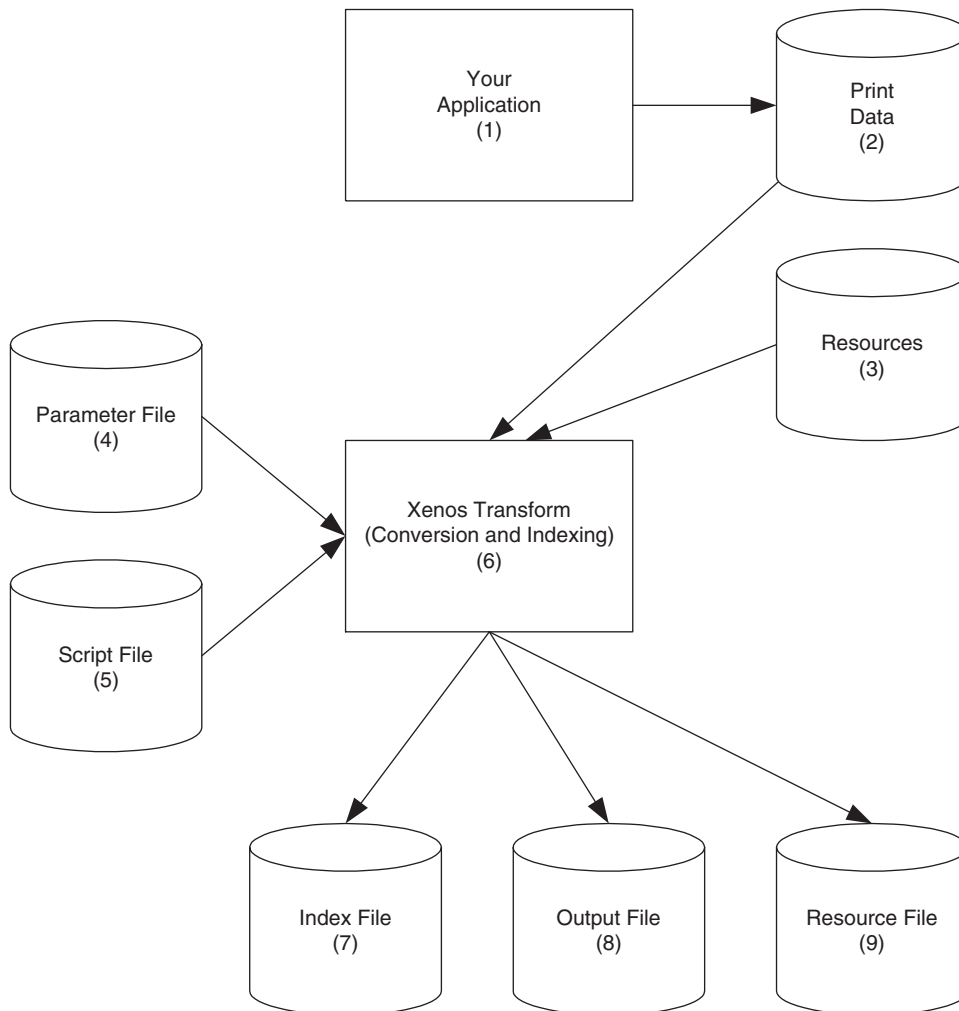


Figure 9. Using the Xenos Transforms to Prepare Files for Loading and Viewing

1. The process begins with your application, which is the program that generates your print data.
2. The print data can be AFP, Metacode, or PCL.
3. Resources are stored in resource libraries.
4. You create the Parameter File, which the Xenos transform uses to locate index data in the input file, convert the input to a specified type of output file, and so forth. See Figure 19 on page 213 for an example of a parameter file.
5. You create the Script File, which the Xenos transform uses to create the OnDemand generic index file and the other output files. See Figure 20 on page 215 for an example of a script file.
6. The ARSLOAD program calls the Xenos transform, to index the print data, convert the input file to the specified type of output, and collect the resources (Metacode output only).

7. The Index File contains the index data that is in the OnDemand generic index format.
8. The Output File contains the indexed groups of pages, also known as documents in OnDemand.
9. The Resource File (Metacode output only) contains the resources that are required to view and print the converted documents.

Chapter 43. Loading data

Figure 10 shows the second part of the load process – the ARSLOAD program processes the files that were created by the Xenos transform.

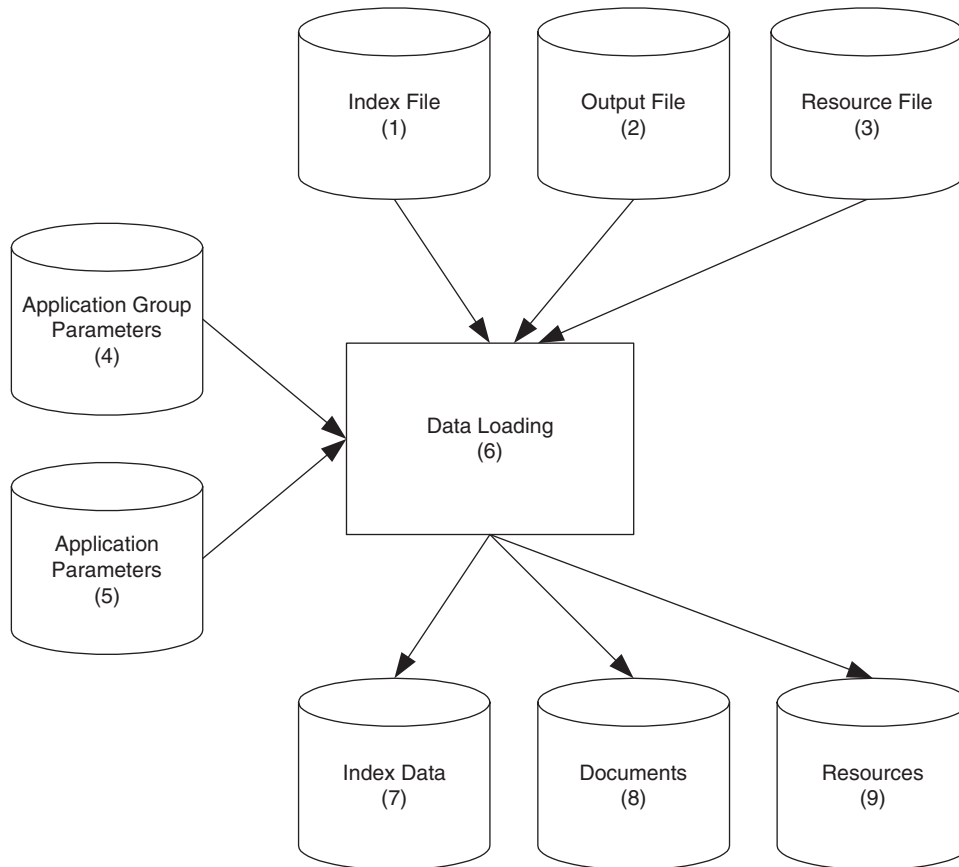


Figure 10. Loading Data into OnDemand

1. The Index File contains the index data that is in the OnDemand generic index format.
2. The Output File contains the indexed groups of pages, also known as documents in OnDemand.
3. The Resource File (Metacode output only) contains the resources that are required to view and print the documents.
4. You create the application group parameters, which include the database and storage management information that the ARSLOAD program uses to process the input files. For example, the database fields that you define for the application group will hold the index field values that the Xenos transform extracted from the original print data.
5. You create the application parameters, which include the type of data and the indexer information that the ARSLOAD program uses to process the input files. For example, on the View Information page in the application, specify the type of data that will be stored in OnDemand, that is, the output from the Xenos transform (AFP, Metacode, or PDF); on the Indexing Information page, specify

Xenos as the indexing program and specify the parameters that are used to process the input file and create the output file, collect the resources, and generate the index data. See “Specifying parameters to OnDemand” on page 198 for information about creating the application parameters. Figure 13 on page 200 and Figure 14 on page 201 show examples of the Indexing Information page.

6. The ARSLOAD program stores the index data into the database and loads the documents and resources on storage volumes.
7. The Index Data is loaded into the database.
8. The Documents are loaded on to storage volumes.
9. The Resources (Metacode output only) are loaded on to storage volumes.

Chapter 44. Scenarios for using Xenos

You can use the Xenos transform to process your files for:

- Viewing and printing locally with the Windows client
- Viewing and printing locally with ODWEK

Viewing and printing with the Windows client

Figure 11 on page 193 shows the steps that you take to view and print documents locally with the Windows client.

1. The process begins with the Index Data (1a) and Documents (1b) that you loaded into the database and on to storage volumes with the ARSLOAD program (AFP or PDF documents).
2. Using the Windows client program, your users search for and view the indexed documents. They can also print documents on local printers from the Windows client.

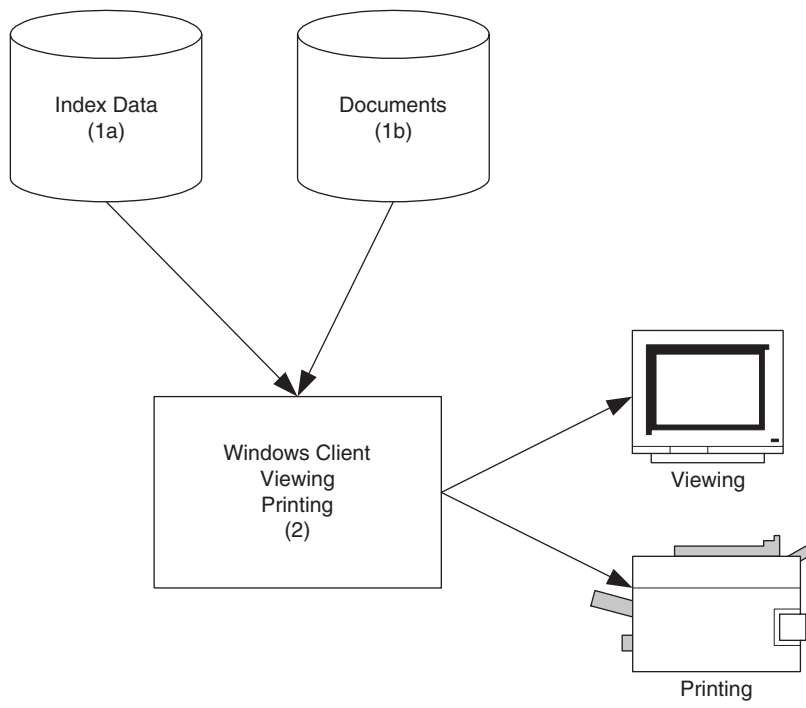


Figure 11. Viewing and Printing with the Windows Client

Viewing and printing with the Web Enablement Kit

Figure 12 on page 195 shows the steps that you take to view and print documents locally with ODWEK.

1. The process begins with the Index Data (1a), Documents (1b), and Resources (1c) that you loaded into the database and on to storage volumes with the ARSLOAD program (AFP, Metacode, or PDF documents).
2. In the ARSWWW.INI file, you specify information about the documents that ODWEK retrieves from the OnDemand server. For AFP and Metacode documents, you can specify that a different type of output should be sent to the Web browser. For example, you can specify that Metacode documents that are retrieved from the system should be transformed into PDF files that are sent to the Web browser. See “Configuring the ARSWWW.INI file” on page 221 for information about how to configure the ARSWWW.INI file to support the Xenos transform.
3. In the ARSXENOS.INI file, you specify the output type and the name of the parameter file and the script file that are used by the Xenos transform to process the input document and create the output that is sent to the Web browser. See “Configuring the ARSXENOS.INI file” on page 223 for an example of the ARSXENOS.INI file.
4. When ODWEK retrieves a document from the system, it checks the ARSWWW.INI file to determine if document conversion is required.
5. If document conversion is required, then ODWEK sends the document to the Xenos transform. Otherwise, ODWEK sends the document to the Web browser.
6. You create the Parameter File, which provides information about the type of conversion that the Xenos transform performs. See “Example of a parameter file” on page 225 for an example of a parameter file.
7. You create the Script file, which the Xenos transform uses to create the output file. See “Example of a script file” on page 227 for an example of a script file.
8. The Xenos transform converts the document from the format in which it was stored on the server to a format that can be viewed at the Web browser.
9. The user views the document from the Web browser. The user can also print documents on local printers from the Web browser.

See Chapter 47, “Configuring ODWEK,” on page 221 for more information about configuring ODWEK to use the Xenos transform.

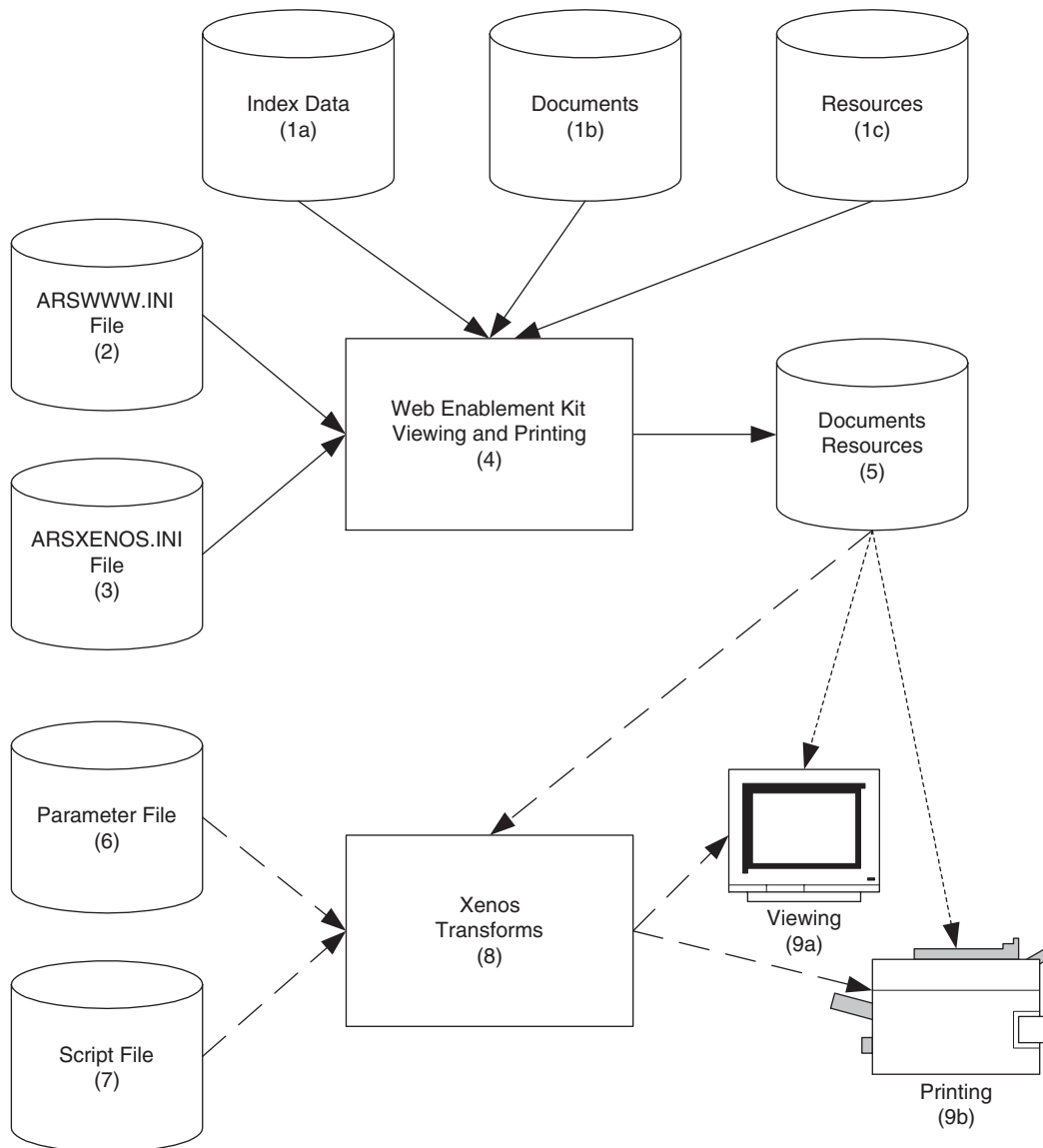


Figure 12. Viewing and Printing with the Web Enablement Kit

Chapter 45. How to specify parameters to the Xenos transforms

This section describes how to specify parameters to the Xenos transforms to assist you with developing indexing parameters and to specify the parameters that are used by the ARSLOAD program and the Xenos transforms to index and convert input files and load them into the system. See Chapter 47, “Configuring ODWEK,” on page 221 for information about how to specify parameters when you use ODWEK to retrieve documents from the system and transform the documents with Xenos before sending them to the Web browser.

There are two parts to specifying the parameters that are used by the ARSLOAD program and the Xenos transform.

- First, you should process some sample input data to determine the locations of the text strings that the Xenos transform uses to identify groups and locate the field index values. You can run the Xenos transform from the command line and specify the parameters for this step.
- Then you should specify the parameters that are used by the ARSLOAD program to call the Xenos transform and specify the parameter file and script file that the Xenos transform uses to process the input data and create the output files. You can specify all of the parameters for this step by using the administrative client.

Processing sample data

You can use the Xenos transform (the JS program) to help you determine the location of the text strings in the input data. The JS program processes one or more pages of an input file and generates an output file. The output file contains one record for each text string on a page. Each record contains the coordinates for a box imposed over the text string (upper left, lower right).

The process works as follows:

- Obtain a printed copy of the sample input file.
- Identify the string values that you want to use to locate fields
- Identify the number of the page on which each string value appears. The number is the *sheet number*, not the page identifier. The sheet number is the order of the page as it appears in the file, beginning with the number 0 (zero), for the first page in the file. A page identifier is user-defined information that identifies each page (for example, iv, 5, and 17-3).
- Process one or more pages of the input file with the JS program.
- In the output file (that is identified by the FDDLOUTPUT parameter), locate the records that contain the string values and make a note of the coordinates.
- Create FIELD parameters using the coordinates, page number, and string value.

See “Generating the locations of text strings on a page” on page 207 for an example of how to run the JS program to process sample data.

For information about the parameters that you can specify, including options and data values, and the script file, see your Xenos documentation.

Specifying parameters to OnDemand

The indexing parameters and other processing parameters that are used by the ARSLOAD program and the Xenos transform are part of the OnDemand application. The administrative client provides an edit window that you can use to maintain the parameters for the application. The parameters are:

- The license file. You must always specify the name of the Xenos license file by using the `OnDemandXenosLicenseFile` parameter. For example:

```
OnDemandXenosLicenseFile=c:\tmp\dmlc_IBM_NT.txt
```

The file that you specify must be a valid license file that you obtained from Xenos.

- The warning level. The `OnDemandXenosWarningLevel` parameter determines how the ARSLOAD program will handle return codes from the Xenos transform. For example:

```
OnDemandXenosWarningLevel=4
```

The Xenos transform sets a return code after it converts an input file. Use this parameter to specify the maximum return code that the ARSLOAD program will consider good and continue with the load process. For example, if you specify 4 (four), then the return code that is set by the Xenos transform must be four or less; otherwise, the load will fail. The default value is zero. If you do not specify this parameter, then the Xenos transform must set a return code of zero. Otherwise, the load will fail. See the Xenos documentation for details about return codes.

- The parameter file. You can use one of two methods to specify the parameters: the *file name* method and the *details* method.

With the file name method, you specify the name of the parameter file. The parameter file that you specify must contain all of the indexing and conversion parameters and other parameters that the Xenos transform uses to process the input data. Use the `OnDemandXenosParmFile` parameter to specify the name of the parameter file. For example:

```
OnDemandXenosParmFile=c:\tmp\sample.par
```

Figure 13 on page 200 shows an example of the Indexing Information page when you use the file name method.

With the details method, you must specify all of the parameters in the edit window within the application. Enclose the parameters inside of the `OnDemandXenosParmBegin` and `OnDemandXenosParmEnd` parameters. For example:

```
OnDemandXenosParmBegin
fddmslib = 'c:\program files\xenos\dmsl.lib'
scriptvar = ( 'Parser', 'META' )
scriptvar = ( 'Generator', 'PDF' )
scriptvar = ( 'NumberOfFields', 2 )
.
.
.
OnDemandXenosParmEnd
```

Figure 14 on page 201 shows an example of the Indexing Information page when you use the details method. **Note:** The example is for demonstration purposes only; not all of the parameters are shown in the edit window.

- The script file. As with the parameter file, you can use the file name method or the details method to specify the script. The script file that you specify must

contain all of the code that the Xenos transform uses to generate the OnDemand generic index file and the other output files. Use the `OnDemandXenosScriptFile` parameter to specify the name of the parameter file. For example:

```
OnDemandXenosScriptFile=c:\tmp\sample.dms
```

Figure 13 on page 200 shows an example of the Indexing Information page when you use the file name method.

With the details method, you must specify all of the script statements in the edit window within the application. Enclose the script statements inside of the `OnDemandXenosScriptBegin` and `OnDemandXenosScriptEnd` parameters. For example:

```
OnDemandXenosScriptBegin
TRUE = 1;
FALSE = 0;
call dm_initialize
rc = dm_SetPart( par_h, 'fdinput', inputfile )
.
.
.
OnDemandXenosScriptEnd
```

Figure 14 on page 201 shows an example of the Indexing Information page when you use the details method. **Note:** The example is for demonstration purposes only; not all of the script statements are shown in the edit window.

See “Indexing and converting input data” on page 212 for an example of a parameter file and a script file to convert and index data and generate the OnDemand generic index file and the other output files.

For information about the parameters that you can specify, including options and data values, and the script file, see your Xenos documentation.

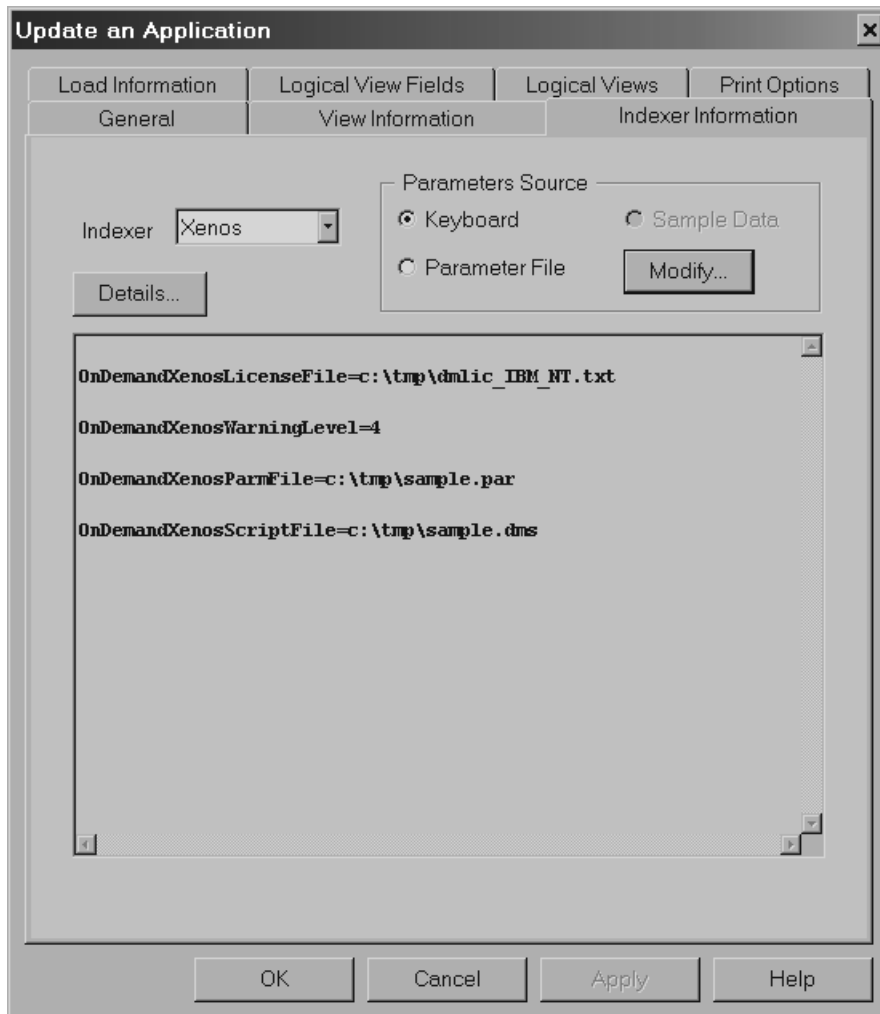


Figure 13. Application Indexing Information – Specifying File Names

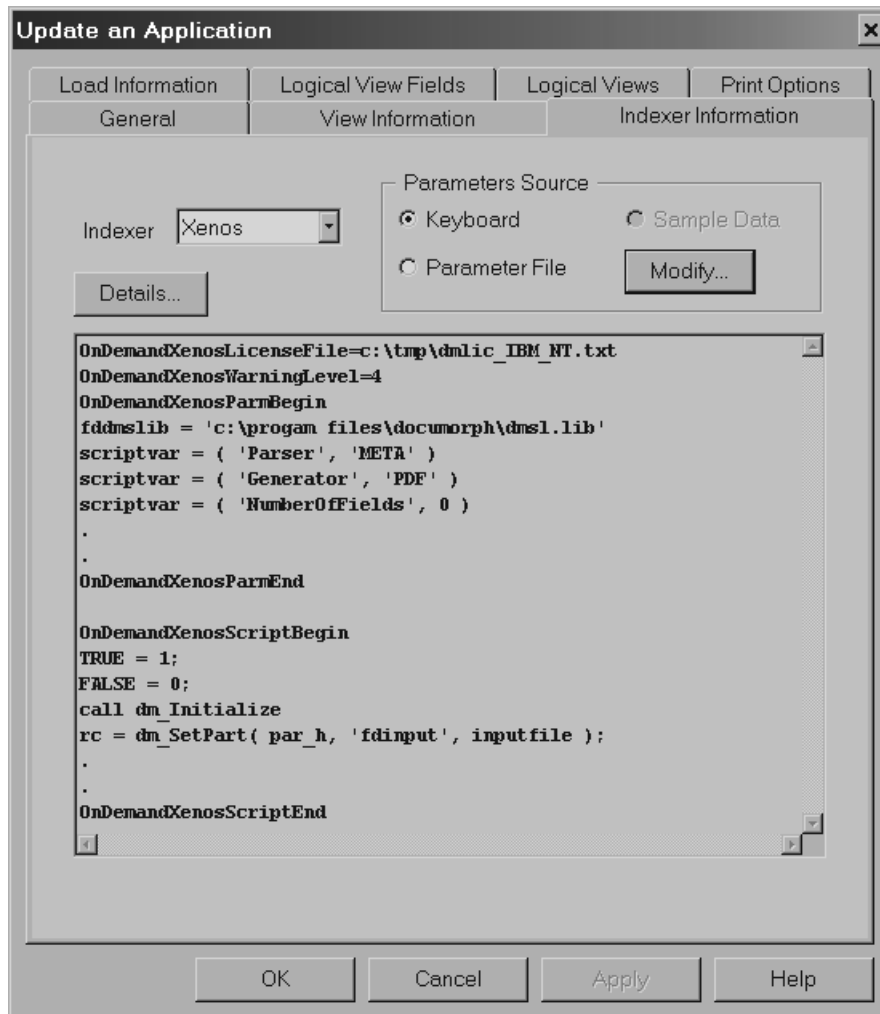


Figure 14. Application Indexing Information – Specifying Details

Chapter 46. JS program reference

Purpose

The JS program is the main Xenos transform program. You can use the JS program in the following ways:

- Manually run the JS program to print the locations of the text strings found on the pages of an input file.
- Automatically call the JS program from the ARSLOAD program to process an input file, generate an OnDemand generic index file and the other output files, convert the input data, and load the data into the system.

The JS program can process input files that contain AFP, Metacode, or PCL data.

Note: See Chapter 47, “Configuring ODWEK,” on page 221 when you use ODWEK to retrieve AFP or Metacode documents from the system and transform them with Xenos before sending the documents to the Web browser.

Syntax

Note: The following shows the syntax that is used by the ARSLOAD program to call the JS program.

Important: All of the parameters are required.

```
▶▶ JS --parms=fileName --report=fileName --scriptvar=inputfile=fileName ▶▶
▶ --scriptvar=outputfile=fileName --scriptvar=indexfile=fileName ▶▶
▶ --scriptvar=resourcefile=fileName --license=fileName ▶▶▶▶
```

Description

The JS program can be used to convert AFP, Metacode, and PCL input data into various types of output formats. The JS program can also extract index data from the input data and collect the resources that are required to view and print the data. The JS program can be used to do the following:

- Print the locations of the text strings found on the pages of an input file. You can use the JS program to help define the indexing fields for your input data. When you define indexing fields, you must specify the location of the string value used to locate the field as a coordinate system imposed on the page. For each string value, you must identify the upper left and lower right position on the page. To help you create the indexing parameters, you can use the JS program to process a sample input file and list the text strings found on the pages of the input file and the locations of the text strings. When you run the JS program to print the locations of the text strings, you must specify the FDDLOUTPUT parameter. The FDDLOUTPUT parameter identifies the name of the file that will contain the text strings and the location information.
- Convert input data, generate an index file, and when generating Metacode output, collect resources. The ARSLOAD program will automatically call the JS

program if the Indexer that is specified on the Indexer Information page under applications is Xenos. You can also run the JS program from the command line to generate an index file manually.

Before you load an input file into the system, you must create an OnDemand application. The application includes the processing parameters that are used by the ARSLOAD program to call the JS program to convert and index the input file and collect resources. When the ARSLOAD program processes an input file and the indexer that is specified on the Indexing Information page in the application is Xenos, the ARSLOAD program automatically calls the JS program to process the input file. The JS program processes the input file with parameters that you specified for the application. Indexing parameters determine the location of the index data on a page. Conversion parameters determine the type of output data that is generated. The JS program extracts the index data from the input file and converts the input data into the specified output format. After indexing and converting the data, OnDemand loads the index data into the database and the indexed groups on to storage volumes.

Parameters

Note: The following shows the parameters that are used by the ARSLOAD program to call the JS program.

Important: All of the parameters are required.

-parms=

Use to specify the file name or full path name of the file that contains the names of the parameter file and the script file. The parameter file contains the parameters that are used to index and convert the input data. The script file creates the OnDemand generic index file and the other output files. See Figure 17 on page 208 and Figure 19 on page 213 for examples of parameter files. See Figure 18 on page 210 and Figure 20 on page 215 for examples of script files. On UNIX servers, file and path names are case sensitive.

Note: For information about the parameters that you can specify, including options and data values, see your Xenos documentation.

-report=

Use to specify the file name or full path name of the file to which a job report will be written. The report will contain a list of the parameters used to process the input file, any error messages, and a list of resources used to process the input file. On UNIX servers, file and path names are case sensitive.

-scriptvar=inputfile=

Use to specify the file name or full path name of the file that contains the input data to process. On UNIX servers, file and path names are case sensitive.

-scriptvar=outputfile=

Use to specify the file name or full path name of the file that will contain the converted output data. On UNIX servers, file and path names are case sensitive.

Note: If you are using the JS program to generate the text strings and locations, then you can discard the output file.

-scriptvar=indexfile=

Use to specify the file name or full path name of the file that will contain the index data that will be loaded into the OnDemand database. On UNIX servers, file and path names are case sensitive.

-scriptvar=resourcefile=

When generating Metacode output, specifies the file name or full path name of the file that will contain the resources that were collected. On UNIX servers, file and path names are case sensitive.

-license=

Use to specify the file name or full path name of the file that contains the license information required to run the Xenos programs. On UNIX servers, file and path names are case sensitive.

Note: You must obtain a valid license file from Xenos.

Examples

The syntax of the JS program is the same, whether you use the program to print the text strings and the location information or you use the program to generate the index and output files to be loaded into OnDemand. The only variables in the process are the type of conversion to run and the indexing parameters that the JS program uses to process the input data, which are specified in the processing parameter file, and the script file.

Figure 15 shows an example of how to run the JS program.

```
js -parms="c:\program files\xenos\parms.auto"
  -report="c:\program files\xenos\sample.rep"
  -scriptvar=inputfile="c:\program files\xenos\sample.mta"
  -scriptvar=outputfile="c:\program files\xenos\sample.out"
  -scriptvar=indexfile="c:\program files\xenos\sample.ind"
  -scriptvar=resourcefile="c:\program files\xenos\sample.res"
  -license="c:\program files\xenos\dmllic_IBM_NT.txt"
```

Figure 15. Running the JS Program

In Figure 15:

- The file that contains the name of the parameter file and the script file is `parms.auto`
- The file that the job report will be written to is `sample.rep`
- The file that contains the input data to process is `sample.mta`
- The file that will contain the converted data to be loaded into the system is `sample.out`
- The file that will contain the index data that will be loaded into the database is `sample.ind`
- The file that will contain the resources that will be loaded into the system is `sample.res`
- The file that contains the Xenos license information is `dmllic_IBM_NT.txt`

Figure 16 on page 206 shows the contents of the example `parms.auto` file.

```
fdjobparm='\program files\xenos\meta2pdf\sample.par'  
fddmscript='\program files\xenos\meta2pdf\sample.dms'
```

Figure 16. Sample Parameter File

In Figure 16, `sample.par` is the name of the file that contains the processing parameters and `sample.dms` is the name of the script file.

Generating the locations of text strings on a page

This example shows how to use the JS program to process a Metacode input file and produce the text strings and location information. You can use the information generated in this step to specify the indexing information that is used by the JS program to extract index data from the input files that you want to load on the system (see “Indexing and converting input data” on page 212).

- Figure 17 on page 208 shows the processing parameters. Note that all of the parameters that are required to process an input file and produce the text strings and the location information are not shown in the example. See your Xenos documentation for a complete list of indexing parameters, options, and data values and more detailed information.

Notes:

1. The script variables Parser and Generator determine the type of conversion taking place. The script variable Parser represents the input file format and the script variable Generator represents the output file format. Table 47 lists the possible values for the Parser and Generator variables when loading input files into the system.

Table 47. Parser and Generator Variables

Input	Output	Parser	Generator
AFP	PDF	scriptvar=('Parser', 'AFP')	scriptvar=('Generator', 'PDF')
Metacode	AFP	scriptvar=('Parser', 'META')	scriptvar=('Generator', 'AFP')
Metacode	Metacode	scriptvar=('Parser', 'META')	scriptvar=('Generator', 'META')
Metacode	PDF	scriptvar=('Parser', 'META')	scriptvar=('Generator', 'PDF')
PCL	PDF	scriptvar=('Parser', 'PCL')	scriptvar=('Generator', 'PDF')

2. The script variable NumberOfFields must be set to 0 (zero) when running the JS program to produce text string coordinates.
 3. The STARTPAGE and STOPPAGE parameters determine how many pages of the input file to process. In the example, the entire input file will be processed. To process a range of pages, specify the number of the starting page to process and the number of pages to process. For example, if you specify STARTPAGE=0 and STOPPAGE=10, then the JS program will process the first eleven pages of the input file. **Note:** The input data must contain the information that the JS program uses to determine when one page ends and another page begins. For example, if the input data contains carriage control characters, then a Skip-to-Channel One carriage control character signals the beginning of a new page.
 4. The FDDLOUTPUT parameter determines the file name or full path name of the file that will contain the text strings and the location information. The FDDLOUTPUT parameter is required when you run the JS to generate the text strings and the locations.
- Figure 18 on page 210 shows the script file. **Note:** The script file does not create an index file, because an index file is not needed for this step.

```

/* Sample processing parameters for Meta2PDF transform */

JS:

/* DM Script Library - XG supplied functions */
fddmslib = 'c:\program files\xenos\dms1.lib'

scriptvar = ('Parser', 'META')
scriptvar = ('Generator', 'PDF')
scriptvar = ('NumberOfFields', 0)

METADL-METAP:

/* Metacode Parser Options */
startjdl = GTIJDJL
startjde = START
startpage = 0
stoppage = 0
position = WORD
native = NO
cc = YES
shade = NONE
dashline = NO
ldmethod = LARGEST
cctran = NONE

/* File Defs */
fdfnt = 'c:\program files\xenos\resources\meta\%s.fnt'
fdjsl = 'c:\program files\xenos\resources\meta\%s.jsl'
respectshift = NO

/* MTA2PDF Options */
fdfonttab = 'c:\program files\xenos\meta2pdf\newfont.tab'

```

Figure 17. Sample Parameters for the Xenos Transform (Part 1 of 2)

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)
```

METADL-DLIST:

```
parmdpi = 300  
pagefilter = all  
resfilter = all  
fddloutput = sample.dl  
  
fieldname = (PAGE)  
fieldword = (20, and, %20)  
fieldphrase = (%500, OneSpace)  
fieldpara = (%500)
```

Figure 17. Sample Parameters for the Xenos Transform (Part 2 of 2)

```

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");

/* open output file */
rc = dm_DASDOpen(dasd_h, '{GROUPFILENAME}'outputfile);

/* initialize */
file_open = FALSE

dlpage = dm_GetDLPage(gen_h);

do while(dlpage <> 'EOF')
  if file_open = FALSE then do
    if(generator = 'AFP') then do
      rc = dm_AFPGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    else if(generator = 'META') then do
      rc = dm_METAGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    else if(generator = 'PDF') then do
      rc = dm_PDFGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    file_open = TRUE
  end
end

```

Figure 18. Sample JS Program Script File (Part 1 of 2)

```

if(generator = 'AFP') then do
  rc = dm_AFPGenWrite(gen_h, dlpage );
end
else if(generator = 'META') then do
  rc = dm_METAGenWrite(gen_h, dlpage );
end
else if(generator = 'PDF') then do
  rc = dm_PDFGenWrite(gen_h, dlpage );
end

dlpage = dm_GetDLPage(par_h);
end

if file_open = TRUE then do
  if(generator = 'AFP') then do
    rc = dm_AFPGenClose( gen_h )
  end
  else if(generator = 'META') then do
    rc = dm_METAGenClose( gen_h )
  end
  else if(generator = 'PDF') then do
    rc = dm_PDFGenClose(gen_h );
  end
end

rc = dm_DASDClose( dasd_h )
return;

```

Figure 18. Sample JS Program Script File (Part 2 of 2)

Indexing and converting input data

This example shows how to use the JS program to process a Metacode input file and produce an index file and a converted output file that can be loaded into OnDemand.

- Figure 19 on page 213 shows the processing parameters. Note that all of the parameters that are required to process an input file are not shown in the example. See your Xenos documentation for a complete list of processing parameters, options, and data values and more detailed information.

Notes:

1. The script variable `NumberOfFields` determines the number of index fields.
 2. The script variables `Field.1` and `Field.2` identify the names and locations of the index fields, using the text string coordinates that were generated in “Generating the locations of text strings on a page” on page 207.
 3. The `fieldlocate` parameter determines the *anchor* character string that the Xenos transform will search for. The Xenos transform will locate the index fields based on the location of the anchor string.
 4. The `fieldbase` parameters identify the locations of index fields using the specified offset from the anchor string.
 5. The `fieldname` parameters identify the names of the index fields.
 6. The `STARTPAGE` and `STOPPAGE` parameters determine how many pages of the input file to process. In the example, the entire input file is processed. **Note:** The input data must contain the information that the JS program uses to determine when one page ends and another page begins. For example, if the input data contains carriage control characters, then a Skip-to-Channel One carriage control character signals the beginning of a new page.
 7. The absence of the `FDDLOUTPUT` parameter means that the JS program will not generate the text strings and the location information.
 8. For information regarding the `FieldLocate`, `FieldName`, and `FieldBase` parameters, refer to your Xenos documentation.
- Figure 20 on page 215 shows the script file. **Note:** The script file inserts a line delimiter at the end of each line that is written to the index file. The example script file uses the line delimiter for Windows systems – `X'0D0A'` (Carriage Return and Line Feed characters). For UNIX systems, the line delimiter is `X'0A'` (Line Feed character).
 - Figure 21 on page 219 shows the OnDemand generic index file that was created by the script file.

```

/* Sample indexing parameters for META2PDF transform */

JS:

/* DM Script Library - XG supplied functions */
fddmslib = 'c:\program files\xenos\dmsl.lib'

scriptvar = ('Parser', 'META')
scriptvar = ('Generator', 'PDF')
scriptvar = ('NumberOfFields', 2)
scriptvar = ('Field.1', 'Name')
scriptvar = ('Field.2', 'Policy')

METADL-METAP:

/* Metacode Parser Options */
startjdl = GTIJDJL
startjde = START
startpage = 0
stoppage = 0
position = WORD
native = NO
cc = YES
shade = NONE
dashline = NO
ldmethod = LARGEST
cctran = NONE

/* File Defs */
fdfnt = 'c:\program files\xenos\resources\meta\%s.fnt'
fdjsl = 'c:\program files\xenos\resources\meta\%s.jsl'
respectshift = NO

```

Figure 19. Sample Parameters for the Xenos Transform (Part 1 of 2)

```

/* MTA2PDF Options */
fdfonttab = 'c:\program files\xenos\meta2pdf\newfont.tab'

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)

METADL-DLIST:

parmdpi     = 300
pagefilter  = all
resfilter   = all

fieldname   = (PAGE)
fieldword   = (20, and, %20)
fieldphrase = (%500, OneSpace)
fieldpara   = (%500)

/* field 1 - extract name */
fieldlocate = ('InsName', 'Insured')
fieldname   = ('Name')
fieldbase   = ('InsName', +275,
              '=' , -35,
              '=' , +800,
              '=' , +30)

/* field 2 - extract policy number */
fieldname   = ('Policy')
fieldbase   = ('InsName', +300,
              '=' , +100,
              '=' , +800,
              '=' , +170)

```

Figure 19. Sample Parameters for the Xenos Transform (Part 2 of 2)


```

TRUE = 1;
FALSE = 0;

call dm_Initialize

par_h = dm_StartParser(Parser);
gen_h = dm_StartGenerator(Generator);

rc = dm_SetParm(par_h, 'dinput', 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 = ""
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"||CrLf);
rc = dm_DASDWrite(index_h, "COMMENT: File generated by the xenos process"||CrLf);
dt = DATE('N');
ts = TIME('N');
rc = dm_DASDWrite(index_h, "COMMENT:" dt||" "||ts||CrLf);
rc = dm_DASDWrite(index_h, "COMMENT:"||CrLf);
rc = dm_DASDWrite(index_h, "CODEPAGE:819"||CrLf||CrLf);

dlpage = dm_GetDLPage(gen_h);

do while(dlpage <> 'EOF')
  if file_open = FALSE then do
    if(generator = 'AFP') then do
      rc = dm_AFPGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    else if(generator = 'META') then do
      rc = dm_METAGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    else if(generator = 'PDF') then do
      rc = dm_PDFGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
  end

```

Figure 20. Sample JS Program Script File (Part 1 of 4)

```

    if rc = 0 then do
        file_open = TRUE
    end
    do i = 1 to NumberOfFields
        fieldvalue.i = dm_GetText( dpage, 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
            docbreak = 1
        end
    end
end
end
if docbreak = 1 then do
    if(generator = 'AFP') then do
        rc = dm_AFPGenClose( gen_h )
    end
    else if(generator = 'META') then do
        rc = dm_METAGenClose( gen_h )
    end
    else if(generator = 'PDF') then do
        rc = dm_PDFGenClose( 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 || crlf )
        field_value = "GROUP_FIELD_VALUE:" || fieldvaluesave.i
        rc = dm_DASDWrite( index_h, field_value || crlf )
    end
end

```

Figure 20. Sample JS Program Script File (Part 2 of 4)

```

/* 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 || crlf)
group_length = "GROUP_LENGTH:" || length
rc = dm_DASDWrite( index_h, group_length || crlf)
group_filename = "GROUP_FILENAME:" || outputfile
rc = dm_DASDWrite( index_h, group_filename || crlf || crlf)

if(generator = 'AFP') then do
  rc = dm_AFPGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
end
else if(generator = 'META') then do
  rc = dm_METAGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
end
else if(generator = 'PDF') then do
  rc = dm_PDFGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
end
if rc = 0 then do
  file_open = TRUE
end
end

if(generator = 'AFP') then do
  rc = dm_AFPGenWrite(gen_h, dlpage );
end
else if(generator = 'META') then do
  rc = dm_METAGenWrite(gen_h, dlpage );
end
end

```

Figure 20. Sample JS Program Script File (Part 3 of 4)

```

else if(generator = 'PDF') then do
    rc = dm_PDFGenWrite(gen_h, dlpage );
end
dlpage = dm_GetDLPage(par_h);
end

if file_open = TRUE then do
    if(generator = 'AFP') then do
        rc = dm_AFPGenClose( gen_h )
    end
    else if(generator = 'META') then do
        rc = dm_METAGenClose( gen_h )
    end
    else if(generator = 'PDF') then do
        rc = dm_PDFGenClose( 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 || crlf)
    field_value = "GROUP_FIELD_VALUE:" || fieldvaluesave.i
    rc = dm_DASDWrite( index_h, field_value || crlf)
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 || crlf)
group_length = "GROUP_LENGTH:" || length
rc = dm_DASDWrite( index_h, group_length || crlf)
group_filename = "GROUP_FILENAME:" || outputfile
rc = dm_DASDWrite( index_h, group_filename || crlf)

rc = dm_DASDClose( dasd_h )
rc = dm_DASDClose( index_h )
return;

```

Figure 20. Sample JS Program Script File (Part 4 of 4)

```
COMMENT: OnDemand Generic Index File Format
COMMENT: This file has been generated by the xenos process
COMMENT: 20 Feb 2001 14:16:18
COMMENT:
CODEPAGE:819
```

```
GROUP_FIELD_NAME:Name
GROUP_FIELD_VALUE:Ward T. Jones, MD.
GROUP_FIELD_NAME:Policy
GROUP_FIELD_VALUE:0030-334-33
GROUP_OFFSET:0
GROUP_LENGTH:111783
GROUP_FILENAME:\program files\xenos\meta2pdf\sample.out
GROUP_FIELD_NAME:Name
GROUP_FIELD_VALUE:WARD T. JONES, MD
GROUP_FIELD_NAME:Policy
GROUP_FIELD_VALUE:0030-334-33
GROUP_OFFSET:111783
GROUP_LENGTH:339119
GROUP_FILENAME:\program files\xenos\meta2pdf\sample.out
GROUP_FIELD_NAME:Name
GROUP_FIELD_VALUE:Mike R. Smith.
GROUP_FIELD_NAME:Policy
GROUP_FIELD_VALUE:0333-888-45
GROUP_OFFSET:450902
GROUP_LENGTH:106689
GROUP_FILENAME:\program files\xenos\meta2pdf\sample.out
GROUP_FIELD_NAME:Name
GROUP_FIELD_VALUE:MIKE R. SMITH
GROUP_FIELD_NAME:Policy
GROUP_FIELD_VALUE:0333-888-45
GROUP_OFFSET:557591
GROUP_LENGTH:338985
GROUP_FILENAME:\program files\xenos\meta2pdf\sample.out
GROUP_FIELD_NAME:Name
GROUP_FIELD_VALUE:Barbara L. Schuster
GROUP_FIELD_NAME:Policy
GROUP_FIELD_VALUE:4567-001-77
GROUP_OFFSET:896576
GROUP_LENGTH:106705
GROUP_FILENAME:\program files\xenos\meta2pdf\sample.out
GROUP_FIELD_NAME:Name
GROUP_FIELD_VALUE:BARBARA L. SCHUSTER
GROUP_FIELD_NAME:Policy
GROUP_FIELD_VALUE:4567-001-77
GROUP_OFFSET:1003281
GROUP_LENGTH:334110
GROUP_FILENAME:\program files\xenos\meta2pdf\sample.out
```

Figure 21. Generic Index File Produced by the JS Program Script

Chapter 47. Configuring ODWEK

You can use ODWEK and the Xenos transforms to retrieve AFP and Metacode documents from an OnDemand server, transform them, and send the output files to the Web browser. For example, you could use the Xenos transform with the ARSLOAD program to process and load Metacode print files. Then, you could use ODWEK to retrieve a Metacode document from the system, call the Xenos transform to convert the Metacode document and send the converted output to the browser.

This section describes how to configure ODWEK to run the Xenos transform. This section contains the following topics:

- Configuring the ARSWWW.INI file
- Configuring the ARSXENOS.INI file
- Example of a parameter file
- Example of a script file
- Changes to the Retrieve Document API

Configuring the ARSWWW.INI file

You must make the following changes to the ARSWWW.INI file to enable the Xenos transform:

- Add the XENOS section
- Specify an AFP transform option
- Specify a Metacode transform option

[XENOS]

The XENOS section contains the parameters that are used by the Xenos transform. You can run the Xenos transform from ODWEK to convert AFP documents into HTML, PDF, or XML output or Metacode documents into AFP, HTML, PDF, or XML output that can be viewed from a Web browser.

Notes:

1. To convert documents, an administrator must obtain the Xenos transforms and install and configure them on the server. See your IBM representative for more information about the Xenos transforms. An administrator must also provide configuration options for the Xenos transforms. See “Configuring the ARSXENOS.INI file” on page 223 for more information about the configuration file.
2. To convert AFP documents with the Xenos transforms, you must specify the AFPVIEWING=XENOS parameter in the DEFAULT BROWSER section (or other browser sections). See “AFPVIEWING” on page 222 for details. (If you plan to use the Retrieve Document API, then you should specify the _afp=XENOS parameter. See “Changes to the Retrieve Document API” on page 229 for details.)
3. To convert Metacode documents with the Xenos transforms, you must specify the METAVIEWING=XENOS parameter in the DEFAULT BROWSER section (or other browser sections). See “METAVIEWING” on page 223 for details. (If you plan

to use the Retrieve Document API, then you should specify the `_meta=XENOS` parameter. See “Changes to the Retrieve Document API” on page 229 for details.)

4. If a document is stored in OnDemand as a large object, then the value of the `ALLOBJECTS` parameter determines how ODWEK handles the document. See “Configuring the ARSXENOS.INI file” on page 223 for details.

This section has a global scope, and you specify it only once in the ARSWWW.INI file.

This section is optional.

This section can contain the following parameters:

CONFIGFILE

Specifies the configuration file that contains the options used by the Xenos transform to convert AFP and Metacode documents. “Configuring the ARSXENOS.INI file” on page 223 provides information about the configuration file that is provided with OnDemand.

This parameter has a global scope, and you specify it only once in the XENOS section.

This parameter is optional.

Example:

```
[XENOS]
CONFIGFILE=/usr/lpp/ars/www/arsxenos.ini
```

INSTALLDIR

Specifies the directory that contains the Xenos transform programs and configuration files. Specify the full path name of the directory on the Web server.

Note: Verify the permissions of the directory that you specify. The processes that run WEK programs must read the install directory.

This parameter has a global scope, and you specify it only once in the XENOS section.

This parameter is optional.

Example:

```
[XENOS]
INSTALLDIR=/usr/lpp/ars/www
```

Specifying the AFP transform option

If you plan to process AFP documents with the Xenos transform, then you need to set the value of the `AFPVIEWING` parameter in the DEFAULT BROWSER (or other browser sections) of the ARSWWW.INI file.

AFPVIEWING

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 the Web browser. To convert AFP documents to HTML, PDF, or XML output with the Xenos transform, specify `AFPVIEWING=XENOS` so that ODWEK will call the Xenos transform to convert the AFP document before

sending it to the Web browser. The type of output that is generated is determined by the value of the OUTPUTTYPE parameter in the ARSXENOS.INI file (see “Configuring the ARSXENOS.INI file”).

This parameter has a global scope, and you specify it only once in the DEFAULT BROWSER section. When using the Retrieve Document function, you can override the specified action with the `_afp` parameter.

This parameter is optional.

Example:

```
[DEFAULT BROWSER]
AFPVIEWING=XENOS
```

Specifying the Metacode transform option

If you plan to process Metacode documents with the Xenos transform, then you need to set the value of the METAVIEWING parameter in the DEFAULT BROWSER (or other browser sections) of the ARSWWW.INI file.

METAVIEWING

When ODWEK retrieves a Metacode document from the OnDemand server, the value of this parameter determines what action, if any, that ODWEK takes before sending the document to the Web browser. To convert Metacode documents to AFP, HTML, PDF, or XML output with the Xenos transform, specify METAVIEWING=XENOS so that ODWEK will call the Xenos transform to convert the Metacode document before sending it to the Web browser. The type of output that is generated is determined by the value of the OUTPUTTYPE parameter in the ARSXENOS.INI file (see “Configuring the ARSXENOS.INI file”).

You can set the parameter to one of the following values:

- | | |
|---------------|---|
| NATIVE | ODWEK extracts and will uncompress Metacode documents and their resources from OnDemand. |
| XENOS | ODWEK calls the Xenos transform to convert Metacode documents to AFP, HTML, PDF, or XML output. |

This parameter has a global scope, and you specify it only once in the DEFAULT BROWSER section. When using the Retrieve Document function, you can override the specified action with the `_meta` parameter.

This parameter is optional.

Example:

```
[DEFAULT BROWSER]
METAVIEWING=XENOS
```

Configuring the ARSXENOS.INI file

The Xenos transforms can convert AFP documents into HTML, PDF, or XML output or Metacode documents into AFP, HTML, PDF, or XML output that can be viewed from a Web browser. The Xenos transforms are licensed software. An administrator must install and configure the Xenos transforms on the Web server. See your IBM representative for more information about the Xenos transforms. An administrator must also specify configuration options for the documents that you plan to process with the Xenos transforms. This section describes how to specify the configuration options.

Note: In this document, the name ARSXENOS.INI refers to the configuration file. To specify the file that contains the configuration options, see “CONFIGFILE” on page 222.

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 in the ARSXENOS.INI file.

The following topics provide additional information about the ARSXENOS.INI file:

- Specifying the ARSXENOS.INI file
- Viewing converted documents

Note: To convert AFP documents with the Xenos transform, you must also specify the AFPVIEWING=XENOS parameter in the DEFAULT BROWSER section (or other browser sections) of the ARSWWW.INI file. See “AFPVIEWING” on page 222 for details. (If you plan to use the Retrieve Document API, then you should specify the _afp=XENOS parameter. See “Changes to the Retrieve Document API” on page 229 for details.) To convert Metacode documents with the Xenos transform, you must also specify the METAVIEWING=XENOS parameter in the DEFAULT BROWSER section (or other browser sections) of the ARSWWW.INI file. See “METAVIEWING” on page 223 for details. (If you plan to use the Retrieve Document API, then you should specify the _meta=XENOS parameter. See “Changes to the Retrieve Document API” on page 229 for details.)

Specifying the ARSXENOS.INI file

The following is an example of an ARSXENOS.INI file:

```
[CREDIT-CREDIT]
ParmFile=/usr/lpp/ars/www/afp2pdf/sample.par
ScriptFile=/usr/lpp/ars/www/noindex.dms
LicenseFile=/usr/lpp/ars/www/dmlic.txt
OutputType=pdf

[default]
ParmFile=/usr/lpp/ars/www/afp2pdf/sample.par
ScriptFile=/usr/lpp/ars/www/noindex.dms
LicenseFile=/usr/lpp/ars/www/dmlic.txt
OutputType=pdf
AllObjects=0
WarningLevel=4
```

The structure of the file is similar to a Windows INI file, and contains one stanza for each OnDemand application and one default stanza. The title line of the stanza 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 in OnDemand. If the application group contains more than one application, then create one stanza for each application.

The parameters that you specify in the [default] stanza are used by the Xenos transform to process documents from applications that are not identified in the ARSXENOS.INI file. The default parameters are also used if an application stanza does not include one of the parameters.

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. See the Xenos documentation for details about the parameters that you can specify in the parameter file. “Example of a parameter file” shows an example of a parameter file.

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. See the Xenos documentation for details about the script file. “Example of a script file” on page 227 shows an example of a script file.

The `LicenseFile` parameter identifies the full path name of a valid license file that you 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.

The `AllObjects` parameter determines how ODWEK will process documents that are stored as large objects in OnDemand. If you specify 0 (zero), then ODWEK will retrieve only the first segment of a document. If you specify 1 (one), then ODWEK will retrieve all of the segments and convert them before sending the document to the viewer. **Note:** If you enable large object support for very large documents, then your users may experience a significant delay before they can view the document.

The `WarningLevel` parameter determines how ODWEK will handle 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 will consider good and send the converted document to the viewer. For example, if you specify 4 (four), then the return code that is set by the Xenos transform must be four or less; otherwise, ODWEK will not send the converted document to the viewer. The default value is zero. If you do not specify this parameter, then the Xenos transform must set a return code of zero. Otherwise, ODWEK will not send the converted document to the viewer. See the Xenos documentation for details about return codes.

Viewing converted documents

To view AFP documents requires the AFP Web Viewer that is provided with ODWEK.

To view HTML or XML documents requires Internet Explorer Version 4.01 or later or Netscape Version 4.06 or later.

To view PDF documents requires an Adobe PDF viewer.

Example of a parameter file

Figure 22 on page 226 shows an example of a parameter file that is used by the Xenos transform to convert AFP documents that are stored in OnDemand into PDF files that ODWEK sends to the Web browser.

The script variables `Parser` and `Generator` determine the type of conversion taking place. The script variable `Parser` represents the input file format and the script variable `Generator` represents the output file format. Table 48 on page 226 lists the

possible values for the Parser and Generator variables when retrieving documents from the system and transforming them before sending the documents to the Web browser.

Table 48. Parser and Generator Variables

Input	Output	Parser	Generator
AFP	HTML	scriptvar=('Parser', 'AFP')	scriptvar=('Generator', 'HTML')
AFP	PDF	scriptvar=('Parser', 'AFP')	scriptvar=('Generator', 'PDF')
Metacode	AFP	scriptvar=('Parser', 'META')	scriptvar=('Generator', 'AFP')
Metacode	HTML	scriptvar=('Parser', 'META')	scriptvar=('Generator', 'HTML')
Metacode	PDF	scriptvar=('Parser', 'META')	scriptvar=('Generator', 'PDF')

```
/* Sample processing parameters for the Xenos transform */
```

```
JS:
```

```
/* DM Script Library - XG supplied functions */
fddmslib = '/usr/lpp/ars/www/dmsl.lib'
```

```
scriptvar = ('Parser', 'AFP')
scriptvar = ('Generator', 'PDF')
```

```
AFPDL-AFPP:
```

```
/* AFP Parser Options */
formdef = fl1a10111
pagedef = pla06462
CC      = on
trc     = off
startpage = 0
stoppage = 0
native  = no
position = word
```

```
/* File Defs */
```

```
FDpagesegs = '/usr/lpp/ars/www/resources/afp/%s.psg'
FDafpfonts = '/usr/lpp/ars/www/resources/afp/%s.fnt'
FDpagedefs = '/usr/lpp/ars/www/resources/afp/%s.pde'
FDformdefs = '/usr/lpp/ars/www/resources/afp/%s.fde'
FDoverlays = '/usr/lpp/ars/www/resources/afp/%s.ovr'
```

```
FDfontcor = '/usr/lpp/ars/www/newfont.tab'
```

```
PDFGEN-PDFOUT:
```

```
/* PDF Out Generator Options */
offset = (0,0)
scaleby = 100
border = NONE
compress = (NONE,NONE,NONE)
orient = AUTO
pdfauthor = 'Xenos Group'
pdfopenact = '/FitH 800'
bmorder = (AsIs,AsIs,AsIs)
```

Figure 22. Sample Parameters for the Xenos Transform

Example of a script file

Figure 23 shows an example of a script file that is used by the Xenos transform to generate the output file that ODWEK sends to the Web browser. **Note:** The script file does not create an index file, because an index file is not needed for this step.

```
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");

/* open output file */
rc = dm_DASDOpen(dasd_h, '{GROUPFILENAME}'outputfile);

/* initialize */
file_open = FALSE

dlpage = dm_GetDLPage(gen_h);

do while(dlpage <> 'EOF')
  if file_open = FALSE then do
    if(generator = 'AFP') then do
      rc = dm_AFPGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    else if(generator = 'HTML') then do
      rc = dm_HTMLGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    else if(generator = 'PDF') then do
      rc = dm_PDFGenOpen(gen_h, '{GROUPFILEENTRY}'outputfile);
    end
    file_open = TRUE
  end
end
```

Figure 23. Sample JS Program Script File (Part 1 of 2)

```

if(generator = 'AFP') then do
  rc = dm_AFPGenWrite(gen_h, dlpage );
end
else if(generator = 'HTML') then do
  rc = dm_HTMLGenWrite(gen_h, dlpage );
end
else if(generator = 'PDF') then do
  rc = dm_PDFGenWrite(gen_h, dlpage );
end

dlpage = dm_GetDLPage(par_h);
end

if file_open = TRUE then do
  if(generator = 'AFP') then do
    rc = dm_AFPGenClose( gen_h )
  end
  else if(generator = 'HTML') then do
    rc = dm_HTMLGenClose( gen_h )
  end
  else if(generator = 'PDF') then do
    rc = dm_PDFGenClose( gen_h )
  end
end

rc = dm_DASDClose( dasd_h )
return;

```

Figure 23. Sample JS Program Script File (Part 2 of 2)

Changes to the Retrieve Document API

This section describes the changes that were made to the Retrieve Document API to support the Xenos transform.

1. The **_afp** parameter now accepts a value of **XENOS**. When you specify **_afp=XENOS**, ODWEK calls the Xenos transform to convert the AFP document that was retrieved from the system into HTML, PDF, or XML output that will be sent to the Web browser. The type of output that is generated is determined by the value of the **OUTPUTTYPE** parameter in the ARSXENOS.INI file (see “Configuring the ARSXENOS.INI file” on page 223).
2. The **_meta** parameter was added. You can use the **_meta** parameter to specify the transform that should take place when ODWEK retrieves a Metacode document from the system. You can specify one of the following values:

NATIVE ODWEK extracts and will uncompress the Metacode document and its resources from OnDemand.

XENOS ODWEK calls the Xenos transform to convert the Metacode document into AFP, HTML, PDF, or XML output. The type of output that is generated is determined by the value of the **OUTPUTTYPE** parameter in the ARSXENOS.INI file (see “Configuring the ARSXENOS.INI file” on page 223).

Accessibility features

The OnDemand product includes a number of features that make the product more accessible for people with disabilities. These features include:

- Features that facilitate keyboard input and navigation
- Features that enhance display properties
- Options for audio and visual alert cues
- Compatibility with assistive technologies
- Compatibility with accessibility features of the operating system
- Accessible documentation formats

Keyboard input and navigation

Keyboard input

The OnDemand clients can be operated using only the keyboard. Menu items and controls provide access keys that allow users to activate a control or select a menu item directly from the keyboard. These keys are self-documenting, in that the access keys are underlined on the control or menu where they appear.

Keyboard focus

In Windows-based systems, the position of the keyboard focus is highlighted, indicating which area of the window is active and where the user's keystrokes will have an effect.

Features for accessible display

The OnDemand clients have a number of features that enhance the user interface and improve accessibility for users with low vision. These accessibility enhancements include support for high-contrast settings and customizable font properties.

High-contrast mode

The OnDemand clients support the high-contrast-mode option that is provided by the operating system. This feature assists users who require a higher degree of contrast between background and foreground colors.

Font settings

In Windows-based systems, the user can specify display settings that determine the color, size, and font for the text in menus and dialog windows. The OnDemand client allows the user to select the font for the document list.

Non-dependence on color

Users do not need to distinguish between colors in order to use any of the functions in this product.

Alternative alert cues

In Windows-based systems, the SoundSentry feature can be used to provide visual feedback for general application and system alerts such as warning beeps.

Compatibility with assistive technologies

The OnDemand clients are compatible with screen reader applications such as Narrator and Via Voice. The OnDemand clients have the properties required for these accessibility applications to make onscreen information available to blind users.

Accessible documentation

Documentation for the OnDemand product is available in HTML format. This allows users to view documentation according to the display preferences set in their browsers. It also allows the use of screen readers and other assistive technologies.

User exits

User exits provided by OnDemand are specific points in the program where an experienced programmer can specify processing routines to enhance or replace the default OnDemand functions. For example, the Logon User Exit provides a point on the library server where you can identify and authenticate users that log on to the system.

Programmers require a working knowledge of the tools needed to develop a user exit program. The following list identifies the main skills and tools that are needed:

- Skills
 - C and C++ programming
 - Operating system programming
 - Experience with relational database technology
 - Knowledge of compiling and linking programs in the C, C++, and operating system environment
 - DB2 UDB, Oracle, or SQL Server (if writing your own SQL code)
- Tools
 - IDE
 - C or C++ compiler

If you do not have these skills, please ask your IBM sales representative about a services contract to implement an OnDemand user exit program.

Notices

Copyright statement

IBM Content Manager OnDemand for Multiplatforms V7.1
AIX, HP-UX, Sun Solaris, and Windows
Release 7.1.0.15
June 30, 2003
5697-G34 (C) COPYRIGHT IBM CORPORATION 2001, 2002
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.

Trademarks

AFP, AIX, CICS, CICS/ESA, C Set ++, DB2, DB2 Universal Database, IBM, Infoprint, iSeries, MVS™, OS/2, OS/390, OS/400, Print Services Facility, pSeries, WebSphere, xSeries, and z/OS are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, the Adobe logo, Acrobat, and the Acrobat logo are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

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

Lotus Notes is a trademark of Lotus® Development Corporation in the United States, other countries, or both.

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

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

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



Program Number: 5697-G34