



OnDemand for AIX

Administrator's Reference

Version 2.2



OnDemand for AIX

Administrator's Reference

Version 2.2

OnDemand for AIX is brought to you by the IBM Printing Systems Company.

Thank you for using OnDemand for AIX, including this book. We helped design and develop this version of OnDemand for AIX, working with a team of programmers, technical writers, testers, and others. With input from customers like you, we have worked to solve your document archive and retrieval needs. We want you to be pleased with the results, and we hope you find this book helpful. If you have comments, we want to hear from you; please use the form at the back of this book.

The OnDemand for AIX Product Team

Second Edition (September 1997)

This edition of *IBM OnDemand for AIX: Administrator's Reference* applies to IBM OnDemand for AIX Version 2 Release 2, program number 5622-662, and to all subsequent releases of this product until otherwise indicated in new releases or technical newsletters.

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

IBM does not warrant that the contents of this publication or the accompanying source code examples, whether individually or as one or more groups, will meet your requirements or that the publication or the source code is error-free.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM authorized Dealer, your IBM Marketing Representative, or your IBM Printing Systems Company Representative.

The IBM Printing Systems Company welcomes your comments. For your convenience, a form for reader's comments is provided at the back of this publication. You may send your comments by fax to 1-800-524-1519, by e-mail to printpub@us.ibm.com, or mail your comments to:

INFORMATION DEVELOPMENT
IBM PRINTING SYSTEMS COMPANY
DEPARTMENT H7FE BUILDING 003G
PO BOX 1900
BOULDER CO 80301-9191

Note: Visit our home page at <http://www.printers.ibm.com>.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

However, the following copyright notice protects this documentation under the Copyright laws of the United States and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

© **Copyright International Business Machines Corporation 1996, 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xv
Trademarks and Service Marks	xv
About this publication	xvii
Who should use this publication	xvii
How this publication is organized	xvii
Related IBM products	xvii
PSF/MVS: MVS Download feature	xvii
Product support	xviii
Our use of typefaces	xx
Platform-specific conventions	xx
Related documentation	xx
ADSTAR Distributed Storage Manager for AIX Version 2.1	xx
AIX Version 4	xxi
DATABASE 2 for AIX Version 2	xxi
MVS TCP/IP	xxi
OnDemand Version 2.2	xxii
Print Services Facility for AIX	xxii
Print Services Facility for MVS	xxii
RS/6000	xxii
Transmission Control Protocol/Internet Protocol	xxii
Things we changed for Release 2.2	xxiii
Changes to the technical content	xxiv

Part 1. Loading Data 1

Chapter 1. Loading reports	3
Overview	3
Preparing to load reports into OnDemand	4
Estimating and allocating storage space	4
Defining the OnDemand application group	4
Defining the OnDemand application	4
Accessing reports, index data, and resources	5
Setting up the ars_load command	6
Indexing reports	7
Indexing reports on the RS/6000	7
Indexing reports on the MVS system	8
Processing index data, reports, and resources	10
Processing reports and resources	10
Processing index data	10
Loading index rows in the database	11
Loading storage objects on storage volumes	12
Cache storage	12
Primary storage node	12

Secondary storage node	12
Resources	13
Backing up databases	13
Backup the OnDemand database	13
Backup the ADSM database	14
Protecting the document cache	14
Checking messages and verifying totals	15
Chapter 2. Loading image files	17
Overview	17
Defining the application group	17
Database organization	17
Expiration type	18
Permissions	18
Field definition	18
Defining the application	18
Application group	18
Data format	18
Indexer	19
Data compression	19
Defining the folder	19
Application group	19
Permissions	19
Folder fields	19
Acquiring the images	19
Accessing the image files	20
Creating index data for the image files	20
Setting up the ars_load command	21
Processing the index data and the image files	22
Processing index data	22
Processing the image files	23
Verifying processing	23
Chapter 3. Restarting a load process	25
Chapter 4. Deleting a report	27
Chapter 5. Data download exits and processing	29
Overview	29
Invoking a data download exit program	29

Part 2. Operating an OnDemand Server 31

Chapter 6. Starting and stopping OnDemand	33
Starting OnDemand	33
Starting DB2 for AIX	34
Starting ADSM	34
Starting MVS Download	34

Starting related MVS programs	35
Starting the OnDemand server daemon	35
Starting the OnDemand data loading daemon	35
Stopping OnDemand	36
Stopping the OnDemand server daemon	38
Stopping ADSM	38
Stopping DB2 for AIX	39
Restarting OnDemand	39
Chapter 7. Monitoring OnDemand	41
Errors and alerts	41
The OnDemand system logging facility	42
Searching for and viewing messages	42
Reviewing a CD-ROM creation manifest	43
System log user exit shell script	43
Monitoring processes	44
Displaying OnDemand process information	44
Displaying DB2 for AIX processes	45
Displaying ADSM for AIX processes	46
Monitoring system load	47
Monitoring memory usage	47
Monitoring system paging space	48
Monitoring disks and file systems	48
Monitoring I/O activity	49
Monitoring storage volumes	50
Monitoring the network	51
Chapter 8. Performance and tuning	53
AIX/6000 tuning considerations	53
Base Operating System performance package	53
AIX Performance Toolbox/6000	54
AIX Performance Monitoring and Tuning Guide	55
DB2/6000 tuning considerations	55
Database system monitor	55
Configuration parameters	56
Index statistics	56
ADSM for AIX tuning considerations	56
TCP/IP tuning considerations	56
OnDemand server	56
Chapter 9. Backup and recovery	59
Overview	59
OnDemand server software	59
OnDemand control files	59
The OnDemand database	60
DB2 for AIX table spaces	60
Database backup	61
Example: creating a full offline backup of the database	62

Database logging	62
Database recovery	63
The ADSM for AIX database	65
ADSM recovery log	66
ADSM volume history file	67
ADSM device configuration file	67
ADSM database recovery	67
Archived report files	68
The document cache	69
Reports on archive media	69
Chapter 10. Adding storage volumes to a library	71
Stop OnDemand	71
Label storage volumes	71
Check storage volumes into the library	72
Restart OnDemand	73
Chapter 11. Removing storage volumes from a library	75
Select the storage volumes to remove	75
Check storage volumes out of the library	77
Offline storage of optical platters	78
Chapter 12. Adding storage volumes for tape backup	79
Stop OnDemand	79
Label storage volumes	79
Restart OnDemand	80
Chapter 13. Adding space to the ADSM database and recovery log	81
Define logical volumes	81
Define disk volumes	82
Determine the maximum extension	83
Extend the capacity	84
Display the assigned capacity	84
Chapter 14. Maintaining migrated index data	87
Configuring the system	87
Logging and routing messages	87
Defining a system log user exit	87
Configuring ADSM for AIX	88
Defining storage sets	88
Configuring application groups	88
What happens when a user queries migrated data	89
Message to the user	89
Message to the system log	89
Message to /dev/console	89
Message to the system log user exit	89
Importing the index data	89
Verify temporary work space	90

Verify database storage space	90
Verify database log file space	90
Run the arsadmin command	90
After you import index data	90
Querying data	90
Expiring imported migrated indexes	90
Verify the length of time before migrating indexes to archive media	91
Verify the length of time to keep imported migrated indexes	91

Part 3. Command Reference 93

Chapter 15. Overview	95
How to read a syntax diagram	95

Chapter 16. ars_adsm command reference	99
Purpose	99
Syntax	99
Description	99
Parameters	99
Examples	100
Notes	100
Files	101

Chapter 17. ars_db command reference	103
Purpose	103
Syntax	103
Description	103
Parameters	104
Examples	107
Notes	108
Files	109

Chapter 18. ars_load command reference	111
Purpose	111
Syntax	111
Description	111
Parameters	112
Examples	114
Notes	115
Files	115

Chapter 19. arsadmin command reference	117
Purpose	117
Syntax	117
Description	120
Parameters	122
Examples	126
Files	127

Chapter 20. arsddate command reference	129
Purpose	129
Syntax	129
Description	129
Parameters	129
Examples	130
Files	132
Chapter 21. arsdocget command reference	133
Purpose	133
Syntax	133
Description	133
Parameters	134
Examples	140
Files	143
Chapter 22. arsjedd command reference	145
Purpose	145
Syntax	145
Description	145
Parameters	146
Examples	146
Notes	147
Files	148
Chapter 23. arsmaint command reference	149
Purpose	149
Syntax	149
Description	149
Parameters	150
Examples	153
Notes	153
Files	153
Chapter 24. arsqquery command reference	155
Purpose	155
Syntax	155
Description	155
Parameters	156
Examples	159
Files	160
Chapter 25. arstblsp command reference	161
Purpose	161
Syntax	161
Description	161
Parameters	161
Examples	162

Files	163
Notes	164
Chapter 26. aruser command reference	165
Purpose	165
Syntax	165
Description	166
Parameters	166
Examples	171
Files	173
Chapter 27. arsviiew command reference	175
Purpose	175
Syntax	175
Description	175
Parameters	175
Examples	176
Files	177
Index	179

Figures

1.	The ars_load Command in the inittab File	6
2.	Indexing Reports on the RS/6000	8
3.	Indexing Reports on MVS	9
4.	Processing Index Data and Reports	10
5.	Offline Backup of the OnDemand Database	14
6.	Full Backup of the ADSM Database	14
7.	Output of the ars_load Command	15
8.	Generic Index File	21
9.	ars_load Command	22
10.	Deleting a Report	27
11.	inittab Records for OnDemand Programs	33
12.	Starting DB2 for AIX	34
13.	Starting ADSM for AIX	34
14.	Starting arsjesd	34
15.	Starting the Server Daemon	35
16.	Starting the Data Loading Daemon	36
17.	Closing the OnDemand Database Connection	37
18.	Stopping DB2 for AIX	37
19.	Stopping ADSM for AIX	38
20.	Monitoring OnDemand Processes with the ps Command	45
21.	Monitoring DB2 for AIX Processes with the ps Command	46
22.	Monitoring ADSM for AIX Processes with the ps Command	47
23.	uptime Command	47
24.	uptime Command Output	47
25.	vmstat Command	47
26.	vmstat Command Output	48
27.	lsps Command	48
28.	lsps Command Output	48
29.	df Command	48
30.	df Command Output	49
31.	iostat Command	49
32.	iostat Command Output	50
33.	iostat Continuous Command	50
34.	ADSM for AIX Query Volume Command	50
35.	ADSM for AIX Query Database Command	50
36.	netstat Command	51
37.	netstat Command Output	51
38.	Get Database Configuration Command	55
39.	Get Database Manager Command	55
40.	Get Snapshot Command	55
41.	tar Command	60
42.	Offline Database Backup Command	62
43.	Labeling Storage Volumes	71
44.	Check In Library Volume Command	73
45.	Query Volume Command	76

46.	Query Volume Command Output	76
47.	Query Volume Command Detail	76
48.	Query Volume Command Detail Output	77
49.	Check Out Library Volume Command	78
50.	Labeling Tape Storage Volumes	79

Tables

1.	OnDemand Load ID	15
2.	ADSM for AIX Logical Volumes	81
3.	Date and Time Format Specifications	130

Notices

References in this publication to products or services of IBM do not suggest or imply that IBM will make them available in all countries where IBM does business or that only products or services of IBM may be used. Noninfringing equivalents may be substituted, but the user must verify that such substitutes, unless expressly designated by IBM, work correctly. No license, expressed or implied, to patents or copyrights of IBM is granted by furnishing this document. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information, which has been exchanged, should contact: IBM Corporation, Printing Systems Company Legal Department, Mail Drop 002S, Boulder, Colorado 80301 USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks and Service Marks

The following terms, used in this publication and other books in the OnDemand library, are trademarks, registered trademarks, or service marks of the IBM Corporation in the United States and other countries:

- ADSM/6000
- ADSTAR
- Advanced Function Printing
- AFP
- AFPDS
- AIX
- AIX/6000
- AIXwindows
- DATABASE 2
- DB2
- IBM
- MO:DCA-P
- MVS
- MVS/XA
- MVS/ESA
- OnDemand for AIX
- OS/2
- PS/2
- PSF/2
- PSF for AIX
- PSF/MVS
- RS/6000
- SP2
- System/370

System/390
SNA
VisuallInfo

The following terms, used in this publication and other books in the OnDemand library, are trademarks of other companies as listed:

Acrobat, Acrobat Exchange, Adobe, Adobe Type Manager, ATM, Distiller, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated.

Cygnnet is a registered trademark of Cygnnet Systems, Inc.

Ethernet is a trademark of Xerox Corporation.

Exabyte is a trademark of Exabyte Corporation.

FaxBox is a trademark of DCE Corporation.

Hewlett-Packard, HP-UX, HP VUE, and PCL are trademarks of Hewlett-Packard Company.

iFOR is a registered trademark of Gradient Technologies, Inc.

Microsoft, TrueType, Windows, Windows 95, and Windows NT are registered trademarks of Microsoft Corporation.

Motif is a trademark of Open Software Foundation, Inc.

Netscape and Netscape Navigator are trademarks of Netscape Communications Corporation.

NFS is a trademark of SUN Microsystems Incorporated.

Novell is a trademark of Novell Inc.

PC/TCP is a trademark of FTP Software, Inc.

Post-it is a registered trademark of 3M.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

WinFax Lite is a trademark of Delrina Corporation.

X Windows is a trademark of the Massachusetts Institute of Technology.

Portions of the OnDemand Windows client program contain licensed software from Adobe Systems Incorporated, © Adobe Systems Incorporated 1987-1997. All rights reserved.

Portions of the OnDemand Windows client program contain licensed software from Pixel Translations Incorporated, © Pixel Translations Incorporated 1990, 1997. All rights reserved.

About this publication

This book contains operational and reference information about IBM OnDemand for AIX Version 2.2 (OnDemand). OnDemand requires the DATABASE 2 AIX/6000 (DB2 for AIX) product. If you plan to store reports on archive media, OnDemand requires the ADSTAR Distributed Storage Manager/6000 (ADSM for AIX) product.

Who should use this publication

This reference is of primary interest to administrators of OnDemand servers. While OnDemand can be managed by a single administrator, administrative responsibilities can be divided among any number of people an installation requires. For example, in a distributed processing environment, one administrator can use server commands to maintain the library server while other administrators can use server commands to maintain object servers. Hopefully you will find using this book and administering OnDemand straightforward.

We assume that you are familiar with AIX/6000 user commands such as *cd*, *ls*, *find*, and *grep*; and you know how to edit files and use redirection and pipes.

How this publication is organized

“Related Documentation,” beginning on page xx contains a list of documents you might find helpful when working with OnDemand, AIX/6000, and other components of the system.

“Loading Data,” beginning on page 1 provides information about indexing and storing report files and creating the index data that OnDemand client programs use to search for and retrieve information.

“Operating the Server,” beginning on page 31 describes general server management tasks, such as starting and stopping the server, monitoring processes and resources, and backup and recovery and typical storage management tasks.

“Command Reference,” beginning on page 93 contains detailed information about OnDemand commands.

Related IBM products

PSF/MVS: MVS Download feature

The MVS Download feature selects output data sets from the JES spool and transfers the data using associated JCL parameters to an OnDemand server.

If you download data from an MVS system to an OnDemand server, refer to the *PSF/MVS: MVS Download Guide* for important information about the MVS Download

feature, including instructions about how to install, configure, and operate the MVS Download feature on an MVS system.

Product support

The IBM Support Center maintains current information about IBM OnDemand for AIX, including program temporary fixes (PTFs).

Before you install OnDemand, contact the IBM Support Center or your IBM software service representative to obtain the latest maintenance level of OnDemand.

If problems occur when you run any of the OnDemand programs, you can call the IBM Support Center to obtain software problem and defect support. The phone number for the IBM Support Center is 1-800-237-5511. Please complete the following information before you call the IBM Support Center.

Problem Summary Form

IBM OnDemand for AIX Version 2.2
Product Number 5622-662

Background Information

Current Date and Time:
Problem Date and Time:
AIX/6000 Level:
AIX/6000 PTFs:

Problem Description

Data Captured

Include core dumps, error message numbers and text, list active processes, and so forth.

Our use of typefaces

Throughout this book, words and phrases appear in **Bold**, *Italic*, and other fonts. The following explains our convention when using these fonts.

Bold	Used for paragraphs that call attention to especially relevant information about a topic or command.
<i>Italic</i>	Used to emphasize concepts and terms.
Monospace	Indicates output of commands and programs in examples. Also used for information you are instructed to type.
UPPERCASE	Indicates parameter or command names.
“Quoted phrase”	Used for references to other parts of the book or other books.

Platform-specific conventions

Note: The AIX conventions are included here as an example.

We use the following AIX conventions in this book:

\$	The prompt for the Bourne and Korn shells. All of the examples in this book assume you use one of these two shells.
#	The prompt for the <i>root</i> (or <i>superuser</i>) user.
\	The AIX command continuation character. The \ (backslash) tells AIX to continue reading the command from the next input line.

You will encounter the words *command* (or *menu*) interface, *command line*, *interactive*, and *prompt* in this book. A command or menu interface is a capability provided by AIX (with *SMIT*, for example) and DB2 for AIX to simplify routine administrative tasks. Most menu interfaces are shells that run the same commands that you would type at the command line. Typing at the command line is the process of interacting with the shell prompt. The shell interprets and executes the commands you type at the prompt.

There may be occasions when you find it necessary to make changes to AIX/6000, OnDemand, ADSM for AIX, DB2 for AIX, and other configuration files and scripts. Whenever you modify a file, first make a backup copy of the current “production” file. Edit the file with a text editor that does not substitute characters (for example, the blank and tab characters) and does not truncate lines. When in doubt, use the *vi* editor.

Related documentation

ADSTAR Distributed Storage Manager for AIX Version 2.1

ADSTAR Distributed Storage Manager for AIX (ADSM) provides optical and tape storage management for the data that you archive in OnDemand. The following publications contain information about or related to ADSM concepts, procedures, and

commands. Refer to the *General Information* book for a complete list of ADSM publications.

General Information, GH35-0131
Administrator's Guide, SH35-0134
Administrator's Reference, SH35-0135
Messages, SH35-0133
Device Configuration, SH35-0137

AIX Version 4

The following publications contain information about or related to AIX Version 4 concepts and procedures:

AIX Version 4 Getting Started, SC23-2527
System User's Guide: Operating System and Devices, SC23-2544
System User's Guide: Communications and Networks, SC23-2545
System Management Guide: Operating System and Devices, SC23-2525
System Management Guide: Communications and Networks, SC23-2526
Guide to Printers and Printing, SC23-2783
Common Desktop Environment 1.0: User's Guide, SC23-2793
Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide, SC23-2795
Messages Guide and Reference, SC23-2641
Performance Monitoring and Tuning Guide, SC23-2365
Problem Solving Guide and Reference, SC23-2606

DATABASE 2 for AIX Version 2

OnDemand stores index data and control information in DATABASE 2 for AIX (DB2 for AIX) database tables. The following publications contain information about DB2 for AIX. Refer to the *Information and Concepts Guide* for a complete list of DB2 for AIX publications.

Information and Concepts Guide, S20H-4664
DB2 for AIX Installation and Operation Guide, S20H-4757
Administration Guide, S20H-4580
Command Reference, S20H-4645
Messages Reference, S20H-4808
Problem Determination Guide, S20H-4779
Database System Monitor Guide and Reference, S20H-4871
SQL Reference, S20H-4665

MVS TCP/IP

The following publications contain information about TCP/IP in the MVS environment:

Planning and Customization, SC31-7134
Programmer's Reference, SC31-7135
Messages and Codes, SC31-7132

OnDemand Version 2.2

The following publications contain information about OnDemand Version 2.2:

Introduction and Planning Guide, G544-5281
Installation and Configuration Guide, G544-5280
Administrator's Reference, S544-5293
Indexing Reference, S544-5489
OS/2 Client User's Guide, S544-5278
Windows Client User's Guide, S544-5275
Getting Started with the User Interface for Windows NT and Windows 95,
S544-5469
Getting Started with the Administrator Interface for Windows NT and Windows 95,
S544-5463

Print Services Facility for AIX

OnDemand uses Print Services Facility for AIX (Print Services Facility) to provide server-based print management. The following publications contain information about Print Services Facility. Refer to the *Print Administration* book for a complete list of Print Services Facility publications.

Facts About PSF for AIX, G544-5305
Print Administration, S544-3817
Print Submission, S544-3878

Print Services Facility for MVS

OnDemand uses the MVS Download feature of Print Services Facility for MVS (PSF for MVS) to support automatic transmission of datasets from the MVS/JES spool to a RS/6000. The following publications contain information about PSF for MVS:

MVS Download Guide, G544-5294
System Programming Guide, S544-3672
Application Programming Guide, S544-3673
Messages and Codes, S544-3675

RS/6000

The following publication contains information about the hardware of the RS/6000:

Getting Started: Using RS/6000, GC23-2377

Transmission Control Protocol/Internet Protocol

The following publications contain information about Transmission Control Protocol/Internet Protocol (TCP/IP). OnDemand requires and uses TCP/IP for communications between servers, clients, MVS systems, and remote devices.

IBM TCP/IP: Introduction, GC31-6080
IBM TCP/IP: Tutorial and Technical Reference, GG24-3376
Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture,
Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
Internetworking with TCP/IP Volume II: Implementation and Internals, Douglas E.
Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

Networking Personal Computers with TCP/IP, Craig Hunt, O'Reilly & Associates, Inc., 1995.

Things we changed for Release 2.2

Revisions to the technical content of this book have been made with the use of revision bars to mark the changes.

Revisions to the editorial content of this book have been made without the use of revision bars to mark the changes.

We removed *Introducing OnDemand*. This information can be found in the *Introduction and Planning Guide*.

We moved the following chapters to the *Indexing Reference*:

- Indexing Report Files
- ARSACIF Command Reference
- Format of the Generic Index File
- ACIF User Exits and Attributes of the Input Print File
- ACIF and the IBM AFP Fonts for ASCII Data
- ACIF Helpful Hints
- ACIF Data Stream Information
- Format of the ACIF Index Object File
- Format of the ACIF Output Document File

We moved *Data Download Exits and Processing* from the Appendix to Part 1, *Loading Data*.

We removed the following chapters. This information can be found in the *Installation and Configuration Guide*:

- Maintaining Cache Storage
- Defining Cache File Systems to OnDemand
- Defining Storage Nodes to ADSM
- Implementing a Copy Storage Pool
- Adding a Storage Library to OnDemand
- Maintaining Database Indexes
- Migrating and Expiring Index Data
- Adding Database File Systems
- Changing the Database Configuration
- Increasing the Number and Size of DB2 Log Files

Changes to the technical content

Description	Page
Updated the OnDemand publication list.	xxii
Added a description of changes to this publication.	xxiii
Changed document cache backup recommendation.	13
Removed sample data download exit program. Contact the IBM support center for more information.	29
The ADSM inittab record now requires a state of wait.	33
Changed the order of stopping OnDemand processes when halting the system.	36
Corrected the flags for the tar command when backing up file systems.	60
Added support for DB2 table spaces.	60
Added support for table space backups and managing backup image files with ADSM.	61
Added support for managing DB2 log files with ADSM.	63
Changed document cache backup recommendation.	68
The ars_db command now supports table space backups and the use of ADSM to manage backup image files.	103
Added restrictions about when to run the ars_db command with certain options.	103
For the ars_load command, must specify both —u and —p flags (or neither).	113
Added recommendations about scheduling ars_load and other OnDemand processes.	115
Enhancements to arsdocget command.	133
Added restrictions about when to run the arsmaint command with certain options.	149
Enhancements to arsquery command.	155
Added the arstblsp command for migrating tables from USERSPACE1 to table spaces.	161

Part 1. Loading Data

This part of the book describes how to load data into OnDemand. The OnDemand data loading programs can process many types of report data, including AFP data, System/370 line data, unformatted ASCII data, and Adobe PDF files. The loading programs also support many image formats, including BMP, GIF, JFIF, PCX, and TIFF. The processing can include indexing the data, to create index data that is loaded into the OnDemand database, segmenting and compressing data, and loading data and resources on storage volumes. The OnDemand database is updated with the index information that allows users to search for and retrieve data using OnDemand client programs. OnDemand supports loading data on cache storage volumes and archive media. You can optionally create a backup copy of data stored on archive media.

Topic	Page
Loading reports	3
Loading image files	17
Restarting a load process	25
Deleting a report	27
Data download exits and processing	29

Chapter 1. Loading reports

Overview

You can archive the print output (reports) of existing application programs in OnDemand without changing the print data stream or writing programs to process the data. OnDemand provides programs that can automatically transmit data from MVS systems to the RS/6000, index the report, store the index information in the database, segment and compress the data, and load the compressed segments on storage volumes. After storing the print output in OnDemand, end-users can query, retrieve, and view or print pages of the report using one of the OnDemand client programs.

The OnDemand data indexing and loading programs can process AIX files or datasets that reside on an MVS system. Typically you transfer a report to the RS/6000 and use the OnDemand indexing and data loading programs to process the report. When you index a report, you must make sure the indexing program can access resources required by the report. You can transfer the resources to the RS/6000 or you can mount an MVS resource library using the Networked File System (NFS). Alternatively, you can index the report on the MVS system and then transmit the report, index, and resource files to the RS/6000. You can concatenate the index, resource, and report files into a single file and send the file to an OnDemand server for processing.

You can create up to sixteen index or filter fields for each type of report that you define to OnDemand, providing many ways for end-users to query information contained in a report. The number of index or filter fields that you define depends on the organization of the data in the report. For example, when you index a report that contains logical items, such as policies and statements, you might define index and filter fields for the statement date, customer name, customer number, balance due, transaction number, and amount. When you index a report that contains transaction data, such as a general ledger, you might define index and filter fields for the report date and transaction number. After you determine and define the index fields, OnDemand automatically extracts the index values from a report and stores them in the database.

OnDemand segments and compresses report data into storage objects using application information contained in the database

Depending on the storage management settings for the application group where you want OnDemand to store a particular report, OnDemand can automatically store a copy of the report in cache storage and create backup copies of the report on archive media.

The data loading program writes output messages to the OnDemand system logging facility at the end of each processing run. You can open the System Log folder and review information, such as the run date, the names of the input files, indexing information, and the number of records processed.

The data loading program provides checkpoints so that you can continue processing a previously interrupted run. You can recover or restart the steps that store index rows into the database and write storage objects on storage volumes.

Preparing to load reports into OnDemand

Estimating and allocating storage space

When you configure a system for OnDemand, you calculate the total amount of magnetic, optical, and tape storage required to support the database, DB2 for AIX log files, archived reports, and the ADSM for AIX database and recovery log, along with temporary space required for OnDemand programs. The amount of storage space required is a factor of how much data that you plan to archive in OnDemand, how long you need to maintain the data in OnDemand, the compression ratio achieved, how long you want OnDemand to maintain index data on magnetic storage volumes, and whether you maintain copies of reports in cache storage and on archive media. If you plan to index data on the RS/6000, you must allocate temporary work space for the data indexing program. Temporary work space is also required for the data loading program.

After you determine how much storage space you need and configure your OnDemand system, you organize the physical storage devices into volume groups that support the various components of OnDemand, such as DB2 for AIX, data download, and cache, and prepare storage devices for use by ADSM for AIX.

The *Introduction and Planning Guide* provides information, formulas, and worksheets that can help you calculate your storage requirements. The *Introduction and Planning Guide* also contains a list of the recommended volume groups to support OnDemand. The *Installation and Configuration Guide* describes how to configure storage libraries and devices and define them to ADSM for AIX.

Defining the OnDemand application group

When you archive reports in OnDemand, you load index data in an application group table and store compressed report data into storage objects associated with the application group. As part of defining a report to OnDemand, you specify the application group where you want OnDemand to store the index data and report file. Use the OnDemand administrator interface to define an application group.

If you plan to store reports on archive media, you must assign the application group to a storage set that identifies a storage node in ADSM for AIX. The *Installation and Configuration Guide* provides details about defining ADSM for AIX storage nodes.

Defining the OnDemand application

Typically you define an application for each different type of report you plan to archive in OnDemand. When you create an application, you define information about the report, such as:

- The application group where you plan to assign the application.

- The data format, for example, the data type, compression information, record format, and code page.
- Indexing information, if you plan to index the report file on the RS/6000.
- Load information that determines special processing that OnDemand must perform on the index data before loading the rows into the database

Use the OnDemand administrator interface to define an application.

Accessing reports, index data, and resources

The typical OnDemand system processes reports created by application programs that run on an MVS System/370. You must create index data for a report before you load the report into OnDemand. If you require resources to display and reprint pages of a report, you must also store the resources in OnDemand.

You can use the OnDemand AFP Conversion and Indexing Facility (ACIF) to index report files and collect the required resources into a resource group. ACIF processes the report and produces an output report file to load, an index file for database input, and a resource group file that contains the resources required to view and reprint pages of the report, such as the form definition, page definition, overlays, and page segments.

ACIF can run on an AIX system or an MVS system. If you run ACIF on an AIX system, you must transfer the report to the RS/6000 and provide access to resources required to process the report. If you run ACIF on an MVS system, you must transfer the report, index, and resources to the RS/6000.

You can use several methods to transfer files between the MVS system and the RS/6000 where you plan to process the data. For example:

- You can use the MVS Download feature to transmit a print dataset from the JES Spool to file systems on the RS/6000. The MVS component of MVS Download operates as one or more JES writers. You configure the writers to interpret JCL parameters, such as CLASS and DEST, and route JES spool files to an OnDemand server. You can specify other JCL parameters, such as FORMS and DATASET, and use an exit program to determine the name of the application group and application so that OnDemand correctly processes the data. The `arsjesd` command starts the AIX component of MVS Download and monitors a specific TCP/IP port for data from an MVS system. You can start one or more occurrences of the `arsjesd` command, each monitoring a different port, receiving datasets from different application programs or MVS systems, and storing the data on different file systems. You can configure MVS Download to automatically transmit and receive report files created by all of the application programs that generate input for OnDemand. The *PSF for MVS: MVS Download Guide* describes how to install, configure, and operate MVS Download on the MVS system, lists the supported JCL parameters, and documents the user exit that provides additional job information to OnDemand. Refer to Chapter 5, “Data download exits and processing” on page 29 for more information about using the `arsjesd` command to invoke a shell script to process a downloaded file and the JCL parameters. If you followed the instructions

in the *Installation and Configuration Guide* you configured the *init* facility to start the *arsjesd* command whenever AIX is initialized.

- You can access files that reside on an MVS system with NFS. For example, if you plan to index reports on the RS/6000, you can NFS mount the MVS resource library on the RS/6000 to access the resources required by a report. However, for performance reasons, we do not recommend using NFS to access datasets that contain the report and index data.
- You can write reports to tape. For example, if the system that generates a report resides in a different location than the system where OnDemand processes the data, you can save a copy of a report on a tape volume and process it at a later time.

Setting up the *ars_load* command

The *ars_load* command is the primary OnDemand indexing and data loading program. The *ars_load* command determines if the input file needs to be indexed, and if so, calls the required indexing program. The *ars_load* command then calls programs that process the index data, load index rows into the database, compress report file segments into storage objects, and load the storage objects on storage volumes.

You schedule and run the *ars_load* command each time that you want to store a report in OnDemand. You can either invoke the *ars_load* command from the AIX prompt or run the *ars_load* command as an AIX daemon that periodically checks specific file systems for data. The *ars_load* command can extract the name of the application group from the download file name or you can specify the name of the application group where you want to archive the report file. By default, the *ars_load* command assumes there is only one application contained in an application group. If there is more than one application contained in the application group, you can modify the *ars_load* script file so that the *ars_load* command extracts the name of the application from the download file name or you must specify the name of the application associated with the report. Refer to Chapter 5, “Data download exits and processing” on page 29 for more information about processing a downloaded file before loading the data and changes you may need to make to the *ars_load* command.

Use the *init* facility to start the *ars_load* command as an AIX daemon. The *init* facility ensures that the *ars_load* command starts each time AIX or the RS/6000 is restarted. Each record that *init* processes defines a specific process and contains runtime parameters. First, determine the parameters and options for the *ars_load* command. Then use the *mkitab* command to add a record to the *inittab* file. Figure 1 shows a sample *inittab* record for the *ars_load* command.

```
arsload:once:/usr/lpp/ars/bin/ars_load
-d /arsacif/acif2 -d /arsacif/acif3 -d /arsacif/acif4
-c /arsacif/acif1 -t 1800 -v
```

Figure 1. The *ars_load* Command in the *inittab* File

The example shows how to start the `ars_load` command to check three different data download directories and create temporary index files in a fourth file system. The `ars_load` command checks the data download directories every thirty minutes. Each of the download directories can be used to temporarily store report files for a different application group (or application within an application group). You can open the System Log folder and review messages generated by the `ars_load` command.

Please refer to Chapter 18, “`ars_load` command reference” on page 111 for a complete description of the `ars_load` command, parameters that you can specify, and information about the files that the `ars_load` command can process. If you followed the instructions in the *Installation and Configuration Guide* you configured the `init` facility to start the `ars_load` command whenever AIX is initialized. Consult your AIX document for information about the `init` facility, the `inittab` file, the `mkitab` command, and AIX system initialization.

Indexing reports

You must create index data for a report before you archive it in OnDemand. You can index reports on the RS/6000 where you plan to run the `ars_load` command or on an MVS system.

Indexing reports on the RS/6000

If you index reports on the RS/6000, you must transfer the report to the RS/6000 and provide access to the resources required by the report. Figure 2 on page 8 depicts one method that you can use to index reports on the RS/6000.

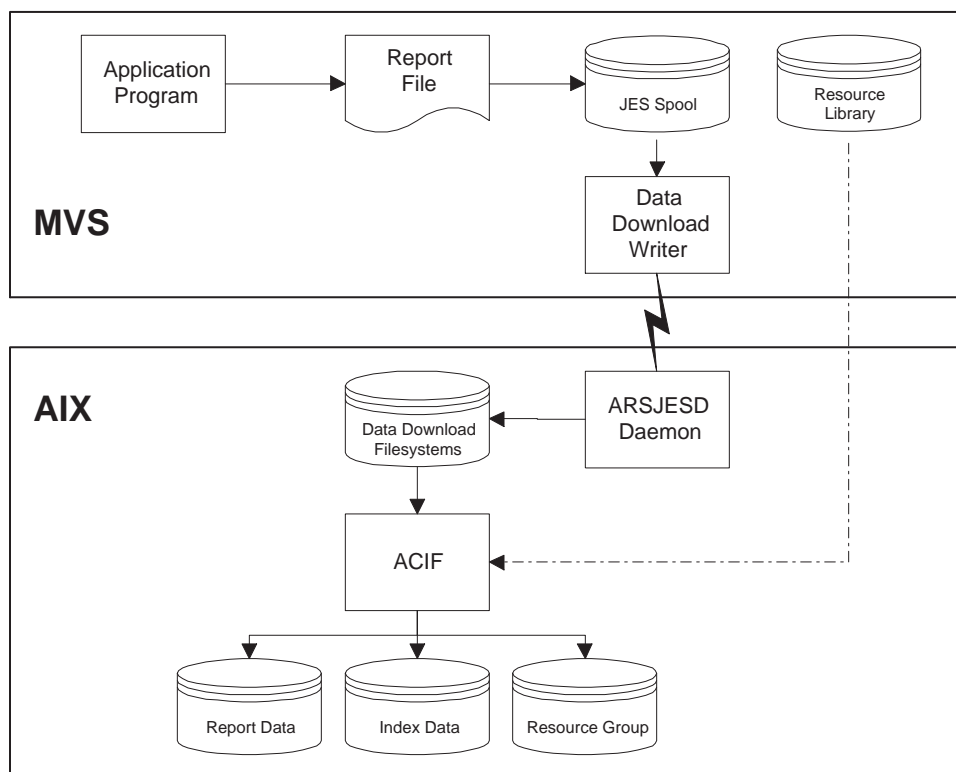


Figure 2. Indexing Reports on the RS/6000

In the example, the MVS Download feature is used to automatically transfer the report to the RS/6000. After the report has been successfully transmitted to the RS/6000, the `ars_load` command can process the data. You can either invoke the `ars_load` command from the AIX prompt or use the `init` facility to start the `ars_load` command to automatically poll the data download directory and process the data as required. Since the report must be indexed before it can be archived in OnDemand, the `ars_load` command automatically calls the data indexing program. In the example, the `ars_load` command calls ACIF to index the report. ACIF accesses the MVS resource library via NFS.

The *Indexing Reference* contains important information about indexing reports, including details about the parameters, options, and data values you can specify to index reports with ACIF.

Indexing reports on the MVS system

If you index reports on an MVS system, you must transfer the report, index file, and resource group file to the RS/6000 where you plan to run the `ars_load` command.

Figure 3 on page 9 depicts one method that you can use to index reports on an MVS system.

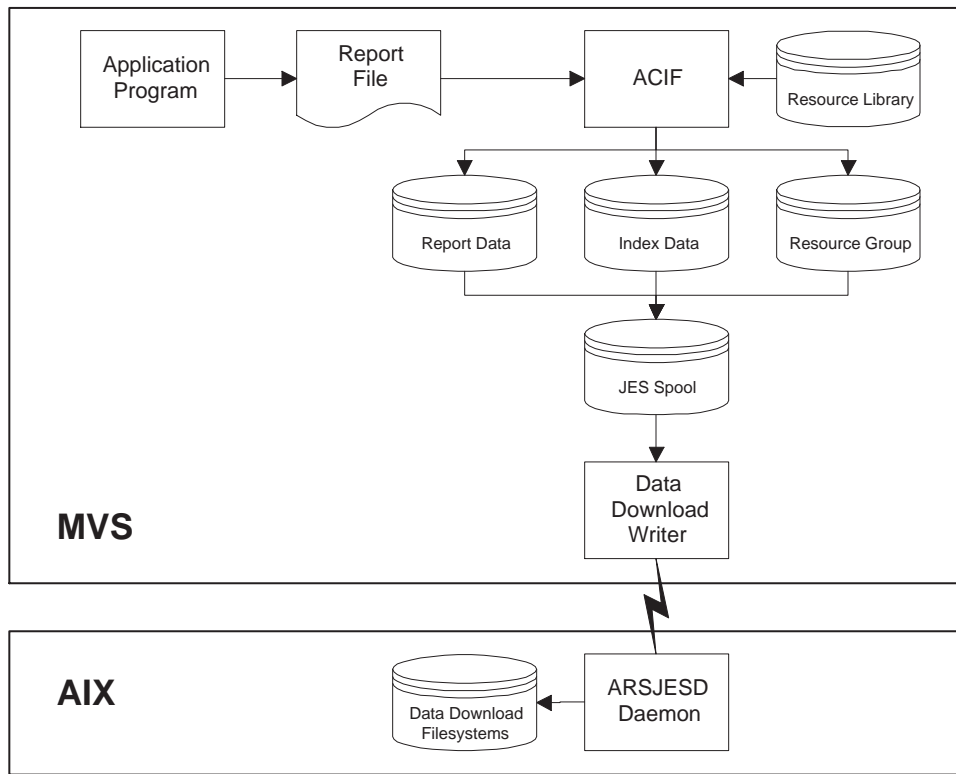


Figure 3. Indexing Reports on MVS

In the example, ACIF is used to index a line data report and convert it to AFP data. ACIF collects the required resources from a library on the MVS system. In this scenario, you must concatenate the three output files created by ACIF into a single file and write the concatenated output file to the JES Spool. MVS Download can be used to automatically transfer the concatenated output file to the RS/6000. After the file has been successfully transmitted to the RS/6000, you must link the concatenated output file (the file type is ARD) to the index file (the file type is IND), the output file (the file type is OUT), and the resource file (the file type is RES) before processing the report with the `ars_load` command. You can create a shell script file that links the files and then invokes the `ars_load` command to process the data. Contact the IBM support center if you have questions about modifying the `ars_load` command.

The *Indexing Reference* provides important information about indexing reports, including details about the parameters, options, and data values that you can specify to index reports with ACIF.

Processing index data, reports, and resources

The `ars_load` command processes the index file to prepare the index data for loading into the application group and processes the report and resource group files to prepare the storage objects that the storage manager loads on storage volumes. Figure 4 shows an overview of the process.

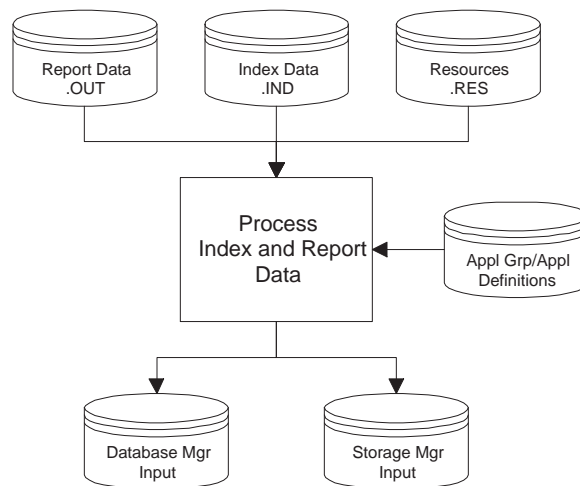


Figure 4. Processing Index Data and Reports

Processing reports and resources

The `ars_load` command segments and compresses report data into storage objects. Segmenting the report improves the efficiency of queries and can improve the time required to retrieve and display the report. Compression improves the efficiency of the storage manager. In this step, the `ars_load` command:

- Reads the application definitions to determine the compression method for the report and the resources.
- Reads the application group storage management definitions.
- Segments the report and compresses the segments into storage objects. OnDemand compresses report data into approximately 100 KB blocks (a default value) and places the blocks into a storage object. OnDemand uses a 10 MB storage object (a default value) to provide better storage efficiency and performance. A compressed object block does not span storage objects. OnDemand assigns unique file names to the storage objects, sequentially numbering the objects within an application group.

Processing index data

The index data created by the indexing program is further processed before it is loaded into the database. The `ars_load` command reads the definitions and parameters that

you specified when you created the application and application group and does the following processing:

- Reads the application group definitions about the database fields.
- Reads the application definitions and does preprocessing, such as mapping index fields to database fields. This step is required if the field names in the index file are not the same as the database field names. You must map the field names so that OnDemand stores index values in the correct database fields.
- After preprocessing the index file, the `ars_load` command creates the database rows:
 - One row for every group of indexed pages in a report file that contains a sorted transaction value.
 - One row for every indexed item in a report file that contains logical items, such as policies and statements.
- Reads the application definitions and does postprocessing, such as dropping duplicate index records. You specify postprocessing parameters when you define the application.
- Passes the index rows to the database manager. The rows consist of fields that contain the index values that the indexing program extracted from the report file and other fields generated by OnDemand. An index row contains:
 - One column for each index and filter field defined in the application group.
 - One or more columns of OnDemand control information.

Loading index rows in the database

The `ars_load` command calls the database manager to load the index rows into an application group table. Depending on the database organization that you selected when you defined the application group, OnDemand either creates a new table for each report file load or adds the rows to an existing database table. OnDemand implements a segment table in DB2 as a high level index to application group data. Rows in the segment table identify the tables of index data for each application group.

Each row in the segment table identifies a different table of application group data. Fields in the segment table contain the application group identifier and a set of dates. The dates represent the earliest and latest date that can be found in that segment of application group data. OnDemand uses the entries in the segment table to limit queries to a specific table or set of application group tables.

When you load a report into OnDemand, the database manager updates the segment table, if the beginning date in the report is earlier than the date already stored in the table.

OnDemand limits the size of an application group table, to provide better performance and storage management. The number of rows in a database table can be set when you define the application group. The default size of a database table is 2.5 million

rows. The database manager automatically closes an application group table and opens a new table when this threshold is reached. The database manager updates the segment table with the latest ending date in the segment of data before closing an application group table.

Loading storage objects on storage volumes

The `ars_load` command calls the storage manager to copy storage objects to cache storage and archive media. The storage manager uses application group information stored in the database to determine where and when to copy storage objects.

Cache storage

Cache storage supports short-term, high-speed retrieval of documents. Cache storage can be located on one or more object servers, depending on how storage sets and primary storage nodes are defined on your system. Documents are always cached on the object server where the primary storage node resides.

If the application group's Length of Time to Cache Data on Magnetic value is greater than zero, the storage manager copies the storage object to cache storage. The Length of Time to Cache Data on Magnetic value determines whether OnDemand caches documents stored in the application group, and if so, for how long.

Primary storage node

The primary storage node provides two functions. It identifies the object server where documents are cached and, when the storage set supports archive media, it identifies an ADSM storage node. ADSM maintains storage objects on archive media for long-term storage.

The storage manager can copy the storage object to archive media when the report is loaded into the system or at a later time. The application group's Length of Time before copying Cache to Archive Media value determines when OnDemand copies the storage object to archive media.

If the primary storage node is cache-only, the storage manager copies the storage object to a cache file system on the object server. If the primary storage node identifies a storage node in ADSM, the storage manager copies the storage object to archive media.

Secondary storage node

A secondary storage node can be used to maintain a copy of files stored on archive media. However, there is no automatic way of accessing the files in a secondary storage node if the files in a primary storage node become corrupted or unavailable. For that reason, if you need to maintain two copies of reports on archive media, we recommend that you implement a copy storage pool in ADSM. Refer to the *Installation and Configuration Guide* for details about implementing a copy storage pool.

If the primary storage node has been assigned to a secondary storage node, the storage manager creates a second copy of the storage object on archive media.

Resources

OnDemand always stores resources in cache storage, providing faster retrieval when a user selects an item for viewing. The `ars_load` command stores only unique resources, checking the resource identifier against resources previously stored in OnDemand.

In addition, if the primary storage node identifies a storage node in ADSM, the storage manager copies the resources to archive media.

Backing up databases

After you store data into OnDemand, we recommend that you create backup images of the OnDemand database and the ADSM database.

- When you backup the OnDemand database, you protect control information and index data that OnDemand and DB2 for AIX require to support the system. Also, after you complete a database backup, DB2 for AIX can reclaim the disk space used by the archived log files.
- When you backup the ADSM database, you protect control information that ADSM for AIX uses to manage report files on optical storage devices.

We recommend that you create backup images of the databases at least once a week, and more often, if you load reports every day. If you followed the instructions in the *Installation and Configuration Guide* you configured the cron facility to automatically start the backup commands on a regular schedule.

Backup the OnDemand database

OnDemand provides the `ars_db` command so that you can create a backup image of the OnDemand database. When you use the `ars_db` command to backup the database and you maintain log files on disk, OnDemand automatically removes log files from the archived log file directory. The `ars_db` command supports two types of database backups, an online backup and an offline backup.

- An online backup can be taken when other applications or processes are connected to the database. That is, other applications and processes can continue to read or modify data while the backup is in progress. An online backup can be scheduled with the cron facility.
- During an offline backup, only the backup task is connected to the database. Before starting an offline backup, we recommend that you stop the OnDemand system to make sure that no other applications or processes are connected to the database.

If your production schedule allows, we strongly encourage you to create an offline backup of the database on a regular schedule, perhaps once a week. Taking an offline backup on a regular schedule can reduce the time required to rebuild the database, if you need to do so.

Figure 5 on page 14 shows an example of how to use the `ars_db` command to create an offline backup of the OnDemand database.

```
/usr/lpp/ars/bin/ars_db -v -y /dev/rmt0
```

Figure 5. Offline Backup of the OnDemand Database

Chapter 9, “Backup and recovery” on page 59 provides more information about backup and recovery strategies and options. Refer to Chapter 17, “ars_db command reference” on page 103 for additional information about how to use the ars_db command to create a backup image of the OnDemand database.

Backup the ADSM database

After you archive data in OnDemand, we recommend that you create a full backup image of the ADSM for AIX database. The backup image can be used to recover the database, in the event a hardware failure occurs or the database becomes corrupted. You can use the ars_adsm command to create a backup image of the ADSM database.

Figure 6 shows an example of how to use the ars_adsm command to backup the database.

```
/usr/lpp/ars/bin/ars_adsm -dv
```

Figure 6. Full Backup of the ADSM Database

In the example, the ars_adsm command writes a backup image of the database to the device specified by the ADSM for AIX dump device class. The *Installation and Configuration Guide* provides details about how to define an ADSM for AIX device class for the database backup. Refer to Chapter 16, “ars_adsm command reference” on page 99 for additional information about how to use the ars_adsm command to create a backup image of the ADSM database.

Protecting the document cache

The document cache is the primary, short-term storage location for reports. If you do not copy reports to archive media when you store them in OnDemand, you need to consider how you can recover the reports in the event you need to do so (for example, if a cache storage device fails).

The document cache can be protected by maintaining it on RAID storage subsystems. RAID storage can provide excellent availability, allowing users to access reports even if a disk or controller fails. The *Introduction and Planning Guide* provides more information about using RAID storage with OnDemand. However, RAID storage is not fail safe. There may be situations when, because of multiple disk or controller failures, users cannot access reports. We encourage you to use ADSM to maintain a copy of reports on archive media.

Checking messages and verifying totals

The `ars_load` command writes output messages to the OnDemand system log at the end of each processing run. You can open the System Log folder and review the messages, such as the run date, the name of the input file, indexing information, and the number of rows loaded into the database. For example:

```
ars_load: Processing file >br1010a<
ars_load: Fri Mar 29 12:37:23 MST 1996 Indexing started,
          394419060 bytes to process
:
ars_load: Fri Mar 29 12:41:33 MST 1996 Indexing completed
ars_load: Fri Mar 29 12:41:35 MST 1996 Load Started,
          394419060 bytes to load
Resource br1010a.res matches the resource 2
OnDemand Load ID = >6850-25-0-15FAA-9577-9577<
Loaded 130000 rows into the database
ars_load: Fri Mar 29 13:05:14 MST 1996 Load Completed
ars_load: Processing successful for file >br1010a<
```

Figure 7. Output of the `ars_load` Command

You can verify the number of records that OnDemand added to the database:

- When processing a report and creating indexed groups of pages, the number of pages in the report divided by the number of pages in an indexed group of pages should equal the number of rows added to the database. For example, if the report contains 150,010 pages and there are 100 pages in an indexed group of pages, OnDemand should have loaded 1,501 rows into the database.
- For a report that contains logical items, such as statements and policies, the number of rows loaded into the database should equal the number of indexed items in the report. For example, if the report contains 1,000 statements, OnDemand should have loaded 1,000 rows into the database.

The `ars_load` command creates one or more Load IDs while processing the report. The OnDemand Load ID represents the storage objects that the `ars_load` command placed in the application group during the load. The Load ID can be used to identify a specific report load. You can use the `arsadmin` command and specify the Load ID to maintain storage objects, for example, to copy storage objects from optical to cache storage and to delete storage objects from OnDemand. Table 1 describes the fields in the Load ID listed in Figure 7.

Load ID Field	Meaning
6850	Application group identifier
25	Primary node identifier
0	Secondary node identifier

Table 1 (Page 2 of 2). OnDemand Load ID

Load ID Field	Meaning
15FAA	The load identifier within the application group
9577	The earliest date in the report
9577	The latest date in the report

Chapter 2. Loading image files

Overview

OnDemand provides support for archiving and retrieving letters and other types of correspondence. The most straight forward way to archive this type of information is to transform a hardcopy letter into an image file with a scanner and image capture software. In addition to scanning the letter and saving it as an image file, you must create index data for the letter. You can then use OnDemand programs to store the index information in the database and the file on a storage volume. OnDemand end-users can then query, retrieve, and view or print copies of the letter using one of the OnDemand client programs.

This chapter outlines how to use a scanner and image capture software to create the image files. Before capturing and archiving files, we explain how to define an application group to store the letters, including the database fields, an application, that describes the format of the data and other processing characteristics, and a folder, that end-users open to search for and retrieve the files.

OnDemand provides a generic index file format to specify index data for image files. The generic index file takes the place of running a program such as ACIF, to extract index data and perform other sophisticated indexing functions. The index file contains the field names and values and identifies the image files to load into OnDemand. You can create up to sixteen index or filter fields for each file that you archive in OnDemand, providing many ways for end-users to query and retrieve the letters. The number of index or filter fields that you define depend on end-user retrieval requirements. In our example, we defined index fields for the date on the letter, the name of the person that sent the letter, the company name, the subject of the letter, and the date that we scanned the letter into the system.

Defining the application group

When you archive files in OnDemand, you store index data into an application group table and load the files into storage objects associated with the application group. This topic provides information about specific application group properties and settings that we made when we defined an application group to archive letters. Use the OnDemand administrator interface to define an application group.

Database organization

We accepted the default Multiple Loads per Database Table database organization and table size of 2.5 million rows. Each time that we store a letter in the database, OnDemand adds one index record to the application group table.

Expiration type

We set the expiration type to Load. This means that OnDemand deletes a report file load at a time from the application group. Depending on how we load data into the application group, one or more letters may be deleted at a time. That is, if we process a generic index file that contains index data for several letters, OnDemand deletes all of the letters from the application group on the same date.

Permissions

We set the Access permission at the application group level. This means that all users defined to OnDemand can access data stored in the application group.

Field definition

We defined the following database fields, which contain the index data that enable users to locate letters based on different criteria, such as the date of the letter, the name of the person that sent the letter, and the subject of the letter. OnDemand stores index data in the fields when we load a letter into the application group. We specify the name of the generic index file that contains the index field names and values when we run the load command.

sdate	The date we scanned the letter into the system. Defined as a date field.
ldate	The date on the letter. Defined as a date field.
name	The person that sent the letter. Defined as a string field that contains variable length data.
company	The person's company. Defined as a string field that contains variable length data.
subject	The subject of the letter. Defined as a string field that contains variable length data.

Defining the application

You must define an application for each different type of file that you plan to archive in OnDemand. This topic provides information about specific application properties and settings that we made when we defined an application for archiving letters. Use the OnDemand administrator interface to define an application.

Application group

We assigned the application to the application group that we created in “Defining the application group” on page 17 to store the letters and the index data.

Data format

We selected TIFF as the data type. When we save the scanned image of a letter, we must save the file as a TIFF file.

Indexer

We selected Generic as the indexer. You must index image files using the generic index file format supported by OnDemand.

Data compression

We specified None, so that OnDemand does not compress the image file. We did this because we plan to save the image files as compressed TIFF files. If you save scanned files as uncompressed TIFF files, you should set compression to LZW16, so that OnDemand can compress the files for better storage efficiency and retrieval performance.

Defining the folder

You must define a folder so that end-users can search for and retrieve the files that you archive in OnDemand. This topic provides information about specific folder properties and settings that we made when we defined a folder to provide access to letters. Use the OnDemand administrator interface to define a folder.

Application group

We selected the application group that we created in “Defining the application group” on page 17 to store the letters and the index data. When users open the folder, they can access the letters stored in the application group.

Permissions

We set the Access permission at the folder level. That means that all users defined to OnDemand can open the folder.

Folder fields

We defined the following fields, which allow end-users to locate letters based on different criteria, such as the date of the letter, the name of the person that sent the letter, and the subject of the letter.

Scan Date	The date we scanned the letter into the system.
Letter Date	The date on the letter.
From	The person that sent the letter.
Company	The person's company.
Subject	The subject of the letter.

We mapped each of the fields to its corresponding application group field.

Acquiring the images

We scanned each letter using an IBM 3119 PageScanner. We previewed and saved the image files using the IBM OS/2 Image Capture Facility. Each letter was saved to a separate file as a compressed TIFF file.

Settings:

- Paper Size 8.5 inches
- Bi-level image
- 300 DPI
- Compressed TIFF

If you plan to let OnDemand compress the data, save the files as uncompressed TIFF files.

Accessing the image files

The OnDemand data indexing and loading programs run under AIX. Therefore, the files that you want to archive in OnDemand must reside on the RS/6000 or you must provide access to the files.

You can transfer the files to the RS/6000 using one of a number of methods. We used *ftp* to copy the files (binary) from the PC that performed the scanning operation to the RS/6000 where we ran the load command.

You can access files that reside on another system with NFS. For example, we could have NFS-mounted the directory that contained the image files on the RS/6000, to access the files required by the load command.

Creating index data for the image files

You must create index data for files before archiving files in OnDemand. The index data for image files must adhere to the generic index file format provided by OnDemand. Refer to the *Indexing Reference* for details about the generic index file.

We used the *vi* editor to create the index file. The following is a sample generic index file:

```
FIELD NAMES BEGIN:
sdate
ldate
name
company
subject
FIELD NAMES END:
12/18/95
09/01/95
Mr. Earl Hawkins
Soft Products
optical storage devices
letter1.tif
0
0
12/18/95
09/01/95
Hans G. Piker
IBM Corporation
optical storage devices
letter2.tif
0
0
12/18/95
09/16/95
Laurie Unicola
Dove Properties
account balance due
letter3.tif
0
0
12/18/95
10/01/95
George VanLocal
Express American
airline fares
letter4.tif
0
0
```

Figure 8. Generic Index File

We saved the index file as *letters.ind* on the RS/6000 in the directory where we ran the load command.

Setting up the `ars_load` command

The `ars_load` command is the program that OnDemand uses to index and store files. The `ars_load` command determines if the input file needs to be indexed, and if so, calls the required indexing program. The `ars_load` command then calls programs that

process the index data, load index rows into the database, optionally compress the file, segment the file into storage objects, and load the storage objects on storage volumes. Please refer to Chapter 18, “ars_load command reference” on page 111 for a complete description of the ars_load command, parameters that you can specify, and information about the files that the ars_load command can process.

You schedule and run the ars_load command each time that you want to archive a file or set of files in OnDemand. You can either invoke the ars_load command from the AIX prompt or run the ars_load command as an AIX daemon that periodically checks specific file systems for data. To support a low volume scanning operation, we chose to run the ars_load command from the AIX prompt, specifying the required options. Figure 9 shows an example of the ars_load command that we used to load the image files into the database.

```
/usr/lpp/ars/bin/arsload -u admin -p "" -n -g Letters letters
```

Figure 9. ars_load Command

The parameters for the ars_load command include:

-u admin	The name of an OnDemand user that can add documents to the application group. This is typically a user with administrator authority for the application group.
-p ""	The password for the OnDemand user.
-n	Do not delete the input files.
-g Letters	The name of the application group to load.
letters	The filename of the input files.

In our example, the ars_load command expects to find the *letters.ind* file in the current directory. This file contains index information that we created in “Creating index data for the image files” on page 20, which is in the OnDemand generic index format. We copied the image files from the PC to the RS/6000, as binary files with a filetype of *tif*.

Processing the index data and the image files

The ars_load command processes the generic index file, loading the index data into the application group and the image files on storage objects that the storage manager loads on storage volumes.

Processing index data

The ars_load command processes the generic index file before loading the data in the database. The ars_load command reads the definitions and parameters that we specified when we created the application and application group and performs the following processing:

- Reads the application group definitions about the database fields.

- Reads the application definitions and performs preprocessing, such as mapping index fields to database fields. This step is required if the field names in the index file are not the same as the database field names.
- After preprocessing the index file, the `ars_load` command creates one row for every image file referenced in the index file. We created a set of index values for each file.
- Reads the application definitions and performs postprocessing, such as dropping duplicate index records. You can specify postprocessing parameters when you define the application.
- Passes the index rows to the database manager. The rows contain the index values and other fields generated by OnDemand. An index row contains:
 - One column for each index and filter field defined for the application group.
 - One or more columns of OnDemand control information.

Processing the image files

The `ars_load` command processes each image file referenced in the generic index file. The `ars_load` command reads the definitions and parameters that we specified when we created the application group and application and performs the following processing:

- Reads the application definitions to determine the compression method for the files.
- Reads the application group storage management definitions.
- Segments the image files and stores the segments into storage objects. OnDemand uses a 10 MB storage object (the default value) to provide better storage efficiency and performance. OnDemand assigns unique file names to the storage objects, sequentially numbering the objects within an application group.

Verifying processing

The `ars_load` command writes output messages to the system log at the end of each processing run. You can open the System Log folder and review the messages, such as the run date, the name of the input file, indexing information, and the number of rows loaded into the database. For example:

```
ars_load: Processing file >letters<
ars_load: Mon Dec 18 13:04:37 MST 1995 Indexing started,
          116017 bytes to process
OnDemand Load ID = >6927-0-0-1FAA-0-0<
Loaded 4 rows into the database
ars_load: Mon Dec 18 13:05:14 MST 1995 Loading Completed
ars_load: Processing successful for file >letters<
```

You can verify the number of records that OnDemand added to the database. When you load a set of image files, the number of rows loaded into the database should equal the number of files that you indexed in the generic index file.

The OnDemand Load ID represents the storage objects that the `ars_load` command placed in the application group during the report file load. The Load ID can be used to identify a report file load. You can use the `arsadmin` command and specify the Load ID to maintain storage objects, for example, to copy storage objects from optical to cache storage and to delete storage objects from OnDemand.

Chapter 3. Restarting a load process

The `ars_load` command terminates when an unrecoverable error occurs during index, database, or storage manager processing. Termination processing includes setting a return code and writing error messages to the system logging facility. Open the System Log folder and view the message log for the load process to find return codes and messages.

If the `ars_load` command fails during index processing, correct the problem and then restart the load from the beginning.

If an error occurs during processing of the database or the storage manager:

- Determine the problem and resolve the problem.
- If a Load ID is listed in the messages written to the system log, use the `arsadmin` command to unload the data. Refer to Chapter 4, "Deleting a report" on page 27 for instructions about how to delete a report from OnDemand.
- Restart the load from the beginning.

Chapter 4. Deleting a report

You can use the *unload* function of the *arsadmin* command to delete a Load ID from one or more storage locations. A Load ID represents the storage objects that the *ars_load* command placed in an application group during a load process.

When you use the *unload* function, you specify the name of the application group, the Load ID, and optionally, a mask. Obtain the Load ID from the message log generated when you loaded the report. If there is more than one Load ID listed in the message log, use the last Load ID listed. The mask determines the storage locations where the storage manager unloads the storage objects. That is, the cache, primary node, or secondary node. By default, the *unload* function deletes storage objects from all three locations. During *unload* processing, the *arsadmin* command deletes rows of index data from the database and updates the database with segment information.

Figure 10 shows an example of an *arsadmin* command that can be used to delete a report from all storage locations in OnDemand.

```
/usr/lpp/ars/bin/arsadmin unload -g 'Credit'  
                                -L 1220-1-0-8FAA-5538-5673
```

Figure 10. Deleting a Report

Please refer to Chapter 19, “arsadmin command reference” on page 117 for a complete description of the *arsadmin* command, parameters that you can specify, and information about how to maintain storage objects in OnDemand.

Chapter 5. Data download exits and processing

Overview

If you use the MVS Download feature to transmit files from an MVS system to an OnDemand server, you may need to preprocess the files before they can be stored into OnDemand. For example, if you use an MVS report distribution system to place the output of application programs on the MVS spool and then transmit the files to the server, you must preprocess the files before loading them, because OnDemand cannot use the transmitted filename to determine the application group and application to load. In this example, the job name (or step name) in the JCL belongs to the report distribution program, not the application program.

The MVS Download feature supports a user exit (APSUX15) that allows you to provide additional job information to an OnDemand server. The MVS Download feature transmits the job information along with the file from the JES spool to a file system on the RS/6000. The *Print Services Facility MVS Download Guide* provides details about the MVS Download feature and the user exit.

The `arsjesd` command provides the `-x` flag so that you can invoke a user-defined shell script to process the job information, after successfully storing a transmitted file on the RS/6000. Chapter 22, “`arsjesd` command reference” on page 145 provides details about the `arsjesd` command.

Invoking a data download exit program

When the MVS Download feature selects output data from the JES spool for transmission to an OnDemand server, it invokes the APSUX15 user exit program. The user exit program concatenates a string of additional job information to the print parameters that the MVS Download feature transmits to the server. On completion, the user exit program passes the location of the string and the string length to the MVS Download feature. The MVS Download feature transmits the output dataset with associated JCL and the additional job information to OnDemand.

The `arsjesd` command receives the dataset into a file system on the RS/6000. If the `arsjesd` command was started with the `-x` flag, it invokes a script file to process the JCL and additional job information. The filename specified with the `-x` flag can be any user-defined shell script file.

For example, you could define a shell script that parses the additional job information passed by MVS Download feature and the APSUX15 user exit program. The script file could extract the value of the `writer` parameter to identify the application to load. Using this value, the script file could query the database to determine the name of the application group. The shell script file could then rename the transmitted file, using the value of the `writer` parameter and the application group name. The shell script could then call the `ars_load` command to store the report into OnDemand. However, the

| ars_load command must be modified to accept the value of the writer parameter as
| the name of the application to load.

| By using the MVS Download feature user exit program, the -x flag on the arsjesd
| command, a user-defined shell script, and a modified version of the ars_load shell
| script file, each file transmitted by MVS Download can be processed so that the
| ars_load command stores the data in the correct application and application group.

| Please contact the IBM support center for more information about the MVS Download
| feature, data download exits, creating shell script files to process reports before loading
| them, and modifying the ars_load command.

Part 2. Operating an OnDemand Server

This section of the book describes the day-to-day, operational tasks that you may perform, such as starting and stopping the OnDemand server programs and related processes and monitoring important information about the operation of the server and server resources, such as processes and system load. This section provides information about tools that can help you monitor the performance of an OnDemand server and recommends other resources that you can use to help with performance and tuning tasks.

This section also contains information about tasks that you may perform on a more infrequent basis, such as backup and recovery of files and data, maintaining storage volumes in optical libraries, and maintaining migrated index data.

Topic	Page
Starting, stopping, and restarting OnDemand processes	33
Monitoring OnDemand	41
Performance and tuning	53
Backup and recovery	59
Adding storage volumes to a library	71
Removing storage volumes from a library	75
Adding storage volumes for tape backup	79
Adding space to the ADSM database and recovery log	81
Maintaining migrated index data	87

Chapter 6. Starting and stopping OnDemand

Starting OnDemand

When you power-on the RS/6000, AIX can automatically start the OnDemand server daemon and other related OnDemand processes as part of AIX initialization using the init command. The init command reads the records in the `/etc/inittab` file. Each record in the file defines a specific process and contains runtime parameters.

If you followed the instructions in the *Installation and Configuration Guide*, you updated the `/etc/inittab` file with the information that the init program requires to start the various OnDemand processes. AIX/6000 automatically starts DB2 for AIX, ADSM for AIX, optionally, MVS Download and the `ars_load` command, and the OnDemand server daemon during system initialization using the init program. Figure 11 shows examples of `/etc/inittab` records for DB2 for AIX, ADSM for AIX, MVS Download, the `ars_load` command, and the server daemon. The name of the server daemon on the library server is `arssockd`. The name of the server daemon on an object server is `arsobjd`.

```
arsdb2:2:wait:su - archive "-c /usr/lpp/ars/bin/ars_db -gkv"  
    >> /tmp/ars_db.log 2>&1;  
  
arsadsm:2:wait:/usr/lpp/ars/bin/ars_adsm -sv  
    >> /tmp/ars_adsm.log 2>&1;  
  
arsjesd:2:once:/usr/lpp/ars/bin/arsjesd -p 6001  
    -d /arsacif/acif1 -d /arsacif/acif2 >> /tmp/ars_db.log 2>&1;  
  
arsload:2:once:/usr/lpp/ars/bin/ars_load -v -c /arsacif/acif3  
    -d /arsacif/acif1 -d /arsacif/acif2  
  
arssockd:2:once:/usr/lpp/ars/bin/arssockd  
  
arsobjd:2:once:/usr/lpp/ars/bin/arsobjd
```

Figure 11. `inittab` Records for OnDemand Programs

If you followed the instructions in the *Installation and Configuration Guide*, you set up the cron facility to start several other programs automatically on a regular schedule. For example, the database and cache storage maintenance program, the database index maintenance program, and programs that backup the OnDemand and ADSM databases and cache file systems. The *Installation and Configuration Guide* provides details about starting OnDemand maintenance programs using the cron facility.

Starting DB2 for AIX

Under normal operating conditions, you never have to start DB2 for AIX on your own. However, if the need should arise, use the `ars_db` command to restart DB2 for AIX.

First, login to AIX as the `root` user. Then enter the following command at the prompt:

```
/usr/lpp/ars/bin/ars_db -gkv
```

Figure 12. Starting DB2 for AIX

The `ars_db` command is a Korn shell script supplied with OnDemand that can be used for general purpose DB2 for AIX functions. Refer to Chapter 17, “`ars_db` command reference” on page 103 for more information about the `ars_db` command, parameters, and options.

Starting ADSM

Under normal operating conditions, you never have to start ADSM for AIX on your own. However, if the need should arise, use the `ars_adsm` command to start ADSM for AIX.

First, login to AIX as the `root` user. Then enter the following command at the prompt:

```
/usr/lpp/ars/bin/ars_adsm -sv
```

Figure 13. Starting ADSM for AIX

The `ars_adsm` command is a Korn shell script supplied with OnDemand that can be used for general purpose ADSM for AIX functions. Refer to Chapter 16, “`ars_adsm` command reference” on page 99 for more information about the `ars_adsm` command, parameters, and options.

Starting MVS Download

Under normal operating conditions, you never have to start the MVS download daemon on your own. However, if the need should arise, use the `arsjesd` command to start the MVS download daemon.

First, login to AIX as the `root` user. Then enter the command at the prompt. For example:

```
/usr/lpp/ars/bin/arsjesd -p 6001  
-d /arsacif/acif1 -d /arsacif/acif2 2>&1
```

Figure 14. Starting `arsjesd`

Replace the strings `/arsacif/acif1` and `/arsacif/acif2` with the names of the file systems that you created to support data download.

The `arsjesd` command operates as an AIX daemon that monitors a TCP/IP port for datasets transmitted by MVS Download running on an MVS system and stores the data in file systems on the RS/6000. Refer to Chapter 22, “`arsjesd` command reference” on page 145 for more information about using the `arsjesd` command to support MVS Download.

Starting related MVS programs

After you power on the RS/6000 or initialize AIX, complete the following steps on the MVS system to enable communications between the RS/6000 and the MVS system.

1. Vary the communications controller between the MVS system and the RS/6000 online.
2. Recycle TCP/IP.
3. If you use the OnDemand CICS/ESA Client program, enable the MVS TCP/IP port for the Client. Invoke the **CSKE** transaction to enable the port. You must specify the TCP/IP port number for the Client, for example, 3000.
4. If you use the MVS Download feature, start an MVS Download writer. You can use the **\$sprtnnnn** command to start a writer. Replace the string **nnnn** with the name of an MVS Download writer. Refer to the *PSF for MVS: MVS Download Guide* for information about operating MVS Download on the MVS system.

Starting the OnDemand server daemon

Under normal operating conditions, you never have to start the OnDemand server daemon on your own. However, if the need should arise, you can start the server daemon from the AIX prompt.

First, login to AIX as the `root` user. Enter the following command at the prompt to start the server daemon on the library server:

```
/usr/lpp/ars/bin/arssockd
```

Figure 15. Starting the Server Daemon

Replace the string `arssockd` with `arsobjd`, to start the server daemon on an object server.

Starting the OnDemand data loading daemon

Under normal operating conditions, you never have to start the OnDemand data loading daemon on your own. However, if the need should arise, use the `at now` and `ars_load` commands to start the data loading daemon, without associating the process with the current logged in user.

First login to AIX as the `root` user. Then enter the commands at the prompt. For example:

```
at now
/usr/lpp/ars/bin/ars_load -v
-d /arsacif/acif1 -d /arsacif/acif2
-c /arsacif/acif3
^D
```

Figure 16. Starting the Data Loading Daemon

In the example, ^D means to press the Ctrl and D keys at the same time. Replace the strings `/arsacif/acif1`, `/arsacif/acif2`, and `/arsacif/acif3` with the names of the file systems that you defined to support data downloading and indexing.

The `ars_load` command is an OnDemand program that operates as an AIX daemon, periodically checking the specified file systems for data to be indexed and stored in OnDemand. Refer to Chapter 18, “`ars_load` command reference” on page 111 for more information about the `ars_load` command, parameters, and options.

Stopping OnDemand

If you need to power off or reboot the RS/6000, reinitialize AIX/6000, or stop OnDemand operation for any reason, complete the following steps to bring the OnDemand operating environment to an orderly conclusion. Stopping OnDemand in this way allows the key processes, such as DB2 for AIX and ADSM for AIX, to terminate normally, clearing buffers, closing log files, and dismounting storage volumes.

Note

The following procedure can require several minutes to complete. If you need to stop the system immediately, halt AIX with the `shutdown` command. Then power off the RS/6000.

1. Login to AIX as the `root` user.
2. If you use the MVS download facility, make sure that no data download operations are in progress. Then, list the PID number of the `arsjesd` command and cancel the process with the `kill` command.
3. Make sure that no data loading operations are in progress. Then, list the PID number of the `ars_load` command and cancel the process with the `kill` command.
4. Make sure that no database or cache maintenance operations are in progress. Then, list the PID numbers of the `ars_db` command that maintains the database indexes and the `arsmaint` command and cancel them with the `kill` command.
5. On all object servers, list the PID number of the `arsobjd` command and cancel the process with the `kill` command.
6. On the library server, list the PID number of the `arssockd` command and cancel the process with the `kill` command. Cancel the process that has a PPID of 1 (one).

7. On the library server, cancel any child processes of the arsockd command with the kill command.

8. Stop DB2 for AIX.

Complete the following steps.

- Load the DB2 for AIX environment, if it is not already loaded. Enter the following command at the AIX prompt:

```
. /home/archive/sql1lib/db2profile
```

- Start the DB2 for AIX command line processor. Enter the following command at the AIX prompt:

```
db2
```

- Connect to the OnDemand database. Enter the following command at the DB2 prompt:

```
connect to archive
```

- Close the OnDemand database connections. Enter the following command at the DB2 prompt:

```
force application all
```

Figure 17. Closing the OnDemand Database Connection

- Stop the DB2 command line processor. Enter the following command at the DB2 prompt:

```
terminate
```

- Stop DB2 for AIX. Enter the following command at the prompt:

```
/usr/lpp/ars/bin/ars_db -hv
```

Figure 18. Stopping DB2 for AIX

9. Start an ADSM for AIX administrative session.

- Issue the QUERY SESSION command to list any open ADSM for AIX sessions.
 - Use the CANCEL SESSION command to stop any open ADSM for AIX sessions.
 - Issue the QUERY MOUNT command to list any mounted storage volumes.
 - Use the DISMOUNT VOL command to cause ADSM for AIX to remove any mounted storage volumes.
10. Stop the ADSM for AIX administrative session.
 11. Stop ADSM for AIX. Type the following command at the prompt:

```
/usr/lpp/ars/bin/ars_adsm -tv
```

Figure 19. Stopping ADSM for AIX

This example uses the value for the ADSM for AIX administrative user and password set in the *ars_adsm* shell script file.

12. If you use the MVS Download feature, stop any active MVS Download writers on the MVS system. You can use the **\$pprtnnnn** command to stop a writer. Replace the string **nnnn** with the name of an active MVS Download writer. Refer to the *PSF for MVS: MVS Download Guide* for information about stopping MVS Download.
13. If you use the MVS Download feature, stop the MVS Download functional subsystem under JES. You can use the **cancel fssname** command to stop the functional subsystem. Replace the string **fssname** with the name of the functional subsystem for MVS Download. Refer to the *PSF for MVS: MVS Download Guide* for information about stopping MVS Download.
14. If you use the OnDemand CICS/ESA Client program, disable the MVS TCP/IP port for the Client. Invoke the **CSKD** transaction to disable the port.
15. If you use either the MVS Download feature or the CICS/ESA Client, vary the communications controller between the MVS system and the RS/6000 offline.
16. Stop AIX. Issue the **shutdown** command.

You can now power off or reboot the RS/6000 or reinitialize AIX.

Stopping the OnDemand server daemon

Complete steps 5 through 7 on page 37 if you need to stop the OnDemand server daemon without stopping other related processes, such as DB2 for AIX and ADSM for AIX.

Stopping ADSM

Caution

Stopping OnDemand in a distributed system, client/server environment should be done in an orderly, structured manner. We recommend you follow the steps outlined in “Stopping OnDemand” on page 36 to shutdown the system so that the various components of OnDemand remain synchronized.

Complete steps 9 through 11 on page 38, if you need to stop ADSM for AIX without stopping other related processes, such as DB2 for AIX and the OnDemand server daemon.

Stopping DB2 for AIX

Caution

Stopping OnDemand in a distributed system, client/server environment should be done in an orderly, structured manner. We recommend you follow the steps outlined in “Stopping OnDemand” on page 36 to shutdown the system so that the various components of OnDemand remain synchronized.

Complete step 8 on page 37 if you need to stop DB2 for AIX without stopping other related processes, such as ADSM for AIX and the OnDemand server daemon.

Restarting OnDemand

AIX automatically starts OnDemand processes when you power on the system. The `init` command starts the OnDemand server daemon and other required processes, such as DB2 for AIX and ADSM for AIX, using records in the `/etc/inittab` file. The cron facility starts OnDemand maintenance programs using records in the `crontab` file. Refer to “Starting OnDemand” on page 33 for information about AIX initialization processing, starting OnDemand processes, and enabling related MVS programs.

Whenever you restart the RS/6000 after a hardware or software problem and AIX did not complete a normal system shutdown, verify that DB2 for AIX and ADSM for AIX are operating without corruption to the OnDemand database and the ADSM database. This is especially important if the `ars_load` command was running when the system problem occurred. If OnDemand was loading data and the system failed, you may need to perform one or more of the following tasks before allowing end-users to access the system:

- Check the configuration of the OnDemand database and DB2 for AIX. Verify the database is in a consistent state, with no backup or rollforward pending.
- Recover the OnDemand database, if required. Refer to “Database recovery” on page 63 for details.
- Audit the ADSM database. Check the database for consistency and correct any problems that are reported. Refer to “ADSM database recovery” on page 67 for details.

- Recover any load files that were being processed when the system failed. Refer to Chapter 3, "Restarting a load process" on page 25 for details.

Chapter 7. Monitoring OnDemand

You probably want to monitor OnDemand resources, such as processes, memory, disk space, and storage volumes, on a regular basis. You can run the commands described in this chapter and save the information reported to track resource utilization and to establish a performance baseline.

We recommend that you run these commands during key, peak processing periods and periods of little or no load. Armed with these statistics, you can quickly determine how the system is performing compared to the baseline. The performance baseline along with, for example, monthly *snapshots*, can help you measure transaction growth, predict the impact of adding applications and users, and determine when you need to add resources, such as storage volumes.

Even if you don't track these items periodically, OnDemand may issue messages that suggest potential problems with the system. Or end-users may complain of sluggish response to search and retrieval requests.

Regardless, the commands described in this chapter can help you spot potential problems and bottlenecks in the system and give you information to begin corrective action.

Chapter 8, "Performance and tuning" on page 53 describes steps that you can take, tools you can use, and reference material that you might find useful for tuning the performance of an OnDemand system, including AIX/6000, DB2 for AIX, and ADSM for AIX.

Errors and alerts

OnDemand assigns a severity to each message generated by server programs. OnDemand sends messages assigned a severity of *alert* or *error* to the system console device, */dev/console*.

OnDemand provides a system logging facility and a message catalog to help you identify and resolve any alerts and errors that you receive. You can open the System Log folder to display messages recorded in the system logging facility. Refer to "The OnDemand system logging facility" on page 42 for more information. You can search the message catalog to obtain details about messages.

For help with specific error messages, refer to the following publications for a listing of error messages and associated articles:

- *AIX Version 4.1 Messages Guide and Reference*
- *DB2 for AIX Messages Reference*
- *ADSM for AIX Messages*

The OnDemand system logging facility

OnDemand provides a logging facility to help the system administrator track OnDemand activity and monitor the system. OnDemand records messages generated by the various OnDemand programs, when you permit logging for system events, user events, and application group events. You can use one of the OnDemand client programs to search for and filter messages by time stamp, severity, message number, and user name.

Note

Before starting OnDemand for the first time, you must initialize the system logging facility. Refer to the *Installation and Customization Guide* for information about initializing the system logging facility.

Searching for and viewing messages

Log on to OnDemand with an OnDemand client program and open the System Log folder.

You can specify a time stamp to search for and retrieve messages. You can specify other search criteria, for example:

- Log ID. Each time an end-user logs on the server, OnDemand assigns a transaction number to that instance of the client program. All requests issued by that instance of the client program includes the same Log ID.
- User ID. The ID (name) of an OnDemand user.
- Severity. OnDemand assigns a severity to each message: Alert, Error, Warning, Info, and Debug.
- View. Depending on the type of message in the log, you may be able to view related records or other data associated with the message from the System Log folder:
 - OnDemand records messages in the database that were issued by commands, such as the `ars_load` command. You can display the messages created during a load by selecting an item that represents the `ars_load` command and choosing the **View all Selected** command.
 - While most messages do not support related records or other data that can be displayed from the System Log folder, you can write a user exit program to record information about the event. For example, you could write a user exit program that keeps track of the number of bytes transferred during a DocGet event. OnDemand provides a system log user exit so that you can process messages and take the appropriate action. Please see the *Installation and Configuration Guide* for more information about the facilities available at the system log user exit.
 - Other messages in the system log do not provide additional records or other data associated with the event, for example, the Logoff event.

- **Msg Num.** The message number issued by OnDemand.
- **Message.** Enter the text of the message that OnDemand uses to restrict a search. For example, if you type **Login**, OnDemand searches for and displays the messages issued by the Logon to a Server command.

When you choose the **Search** command, OnDemand retrieves the messages from the database that match the search criteria you specified.

Reviewing a CD-ROM creation manifest

You can use the system logging facility to review a manifest created by the CD-ROM mastering software, an extra-charge feature of OnDemand. A CD-ROM creation manifest contains a cover page with control information about the CD-ROM and one or more pages that list the items retrieved from the database and written on the CD-ROM.

To review a CD-ROM creation manifest:

1. Log on to OnDemand with an OnDemand client program and open the System Log folder.
2. Type **CD** in the Message entry field.

Note

If you know the CD-ROM volume number of the manifest that you want to review, enter the number in the Message entry field. For example, you can enter **%171** in the Message entry field to locate the manifest for volume number AOD00000171.

3. Specify a date. For example:
 - If you want to display a manifest for a particular date, select the **Equal** operator for the Time Stamp search field and enter the date in the Time Stamp entry field.
 - If you want to display the manifests created during a range of dates or you do not know the exact creation date, select the **Between** operator for the Time Stamp search field and enter a date range in the Time Stamp entry fields.
4. Choose **Search**. OnDemand displays the manifests that match the search criteria that you specified.
5. Review a manifest by selecting an item from the list and choosing the **View All Selected** command.
6. Use the Next Page and Previous Page commands to scroll through the manifest.

System log user exit shell script

When you permit logging for system, user, or application group events, OnDemand passes program-generated messages to the arsllog shell script in the `/usr/lpp/ars/bin` directory. By default, the script does not perform any function. However, you can replace the script with one that does specific functions, such as check the severity of the message and issue an alert using, for example, NetView.

You can configure OnDemand to provide information to the script by setting the appropriate system, user, and application group logging parameters with the administrator interface.

Please see the *Installation and Configuration Guide* for information about the system logging user exit shell script.

Monitoring processes

You can monitor and display status information about OnDemand, DB2 for AIX, and ADSM for AIX processes. To monitor processes, use the `ps` command. The `ps` command is an AIX system command that returns process status information about active processes to standard output. Key fields in the `ps` command output include UID (user), PID (process), C and PRI (CPU use and execution priority), STIME (the time the process started), TIME (accumulated CPU time), and CMD (the command the process is running).

The following examples show how to use the `ps` and `grep` commands to monitor key processes on the OnDemand server. For more information about the `ps` and `grep` commands, type `man ps` or `man grep` at the AIX prompt or refer to your AIX documentation.

Displaying OnDemand process information

OnDemand programs and commands that may be active on the RS/6000 include the following:

- ars_db** The OnDemand command used to maintain the database. Refer to the Chapter 17, “ars_db command reference” on page 103 for information about starting the `ars_db` command to perform database maintenance.
- arssockd** The main OnDemand program that runs on the library server. If OnDemand client programs cannot connect to the server, make sure that this process is active. Refer to “Starting the OnDemand server daemon” on page 35 for instructions about how to restart the `arssockd` command.

There may be more than one `arssockd` process active, depending on the value of the `ARS_NUM_DBSRVR` parameter in the `ars.cfg` file. Please refer to the *Installation and Configuration Guide* for information about the `ARS_NUM_DBSRVR` parameter.
- arsobjd** The main OnDemand program that runs on an object server. If OnDemand client programs experience problems retrieving reports from the server, make sure that this process is active. Refer to “Starting the OnDemand server daemon” on page 35 for instructions about how to restart the `arsobjd` command.
- ars_load** The data loading daemon. To automatically store data in Application Groups, ensure that this process is active. Refer to “Starting the OnDemand data loading daemon” on page 35 for instructions about how to restart the `ars_load` command. You may see two other commands

associated with this process. The `arsacif` and `arsadmin` commands run with the parent process ID of the `ars_load` process.

- arsacif** The data indexing program, invoked by the `ars_load` command.
- arsadmin** The database and storage manager program, invoked by the `ars_load` command.
- arsjesd** The MVS Download daemon; the AIX component of the MVS Download feature. To automatically receive datasets from MVS Download running on an MVS system, ensure that this process is active. Refer to “Starting MVS Download” on page 34 for instructions about how to restart the `arsjesd` command.
- arsmaint** The process that maintains cache storage and the OnDemand database.

Use the following command to display process status:

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

Figure 20. Monitoring OnDemand Processes with the `ps` Command

The output of the command issued on the library server should be similar to the following example:

```
UID  PID  PPID  C  PRI  STIME  TTY  TIME  CMD
root 6022 28991 0 60 13:30:13 - 0:00 arssockd: db (USERID)
root 6022 28991 0 60 13:30:13 - 0:00 arssockd: db (USERID - idle)
root 6018 28991 0 60 13:30:13 - 0:00 arssockd: db (idle)
root 6014 28991 0 60 13:30:13 - 0:00 arssockd: (accepting)
```

Displaying DB2 for AIX processes

DB2 Version 2.1 programs that may be active on the library server include the following:

- db2dlock** The data lock manager; looks for and resolves deadlocks on the database.
- db2licd** The license daemon.
- db2agent** Handles all SQL processing for application requests. Several occurrences of this process may be active.
- db2gds** Responsible for spawning most DB2 daemons.
- db2wdog** The “watch dog” monitors the other processes and cleans up resources after abnormal terminations.
- db2loggr** This process manages the database log files.
- db2sysc** This process handles DB2 startup and shutdown.

- db2pfchr** Allows for more efficient processing via read-ahead, big-block and parallel I/O. Several occurrences of this process may be active.
- db2pclnr** Asynchronously writes dirty pages when the CPU would otherwise be idle. Several occurrences of this process may be active.
- db2ipccm** Handles connection requests from local clients.
- db2resyn** Handles recovery of indoubt transactions.

Refer to “Starting DB2 for AIX” on page 34 for instructions about how to restart DB2 for AIX.

Use the following command to display process status:

```
ps -ef | grep db2
```

Figure 21. Monitoring DB2 for AIX Processes with the ps Command

The output of the command should be similar to the following example:

```

  UID  PID  PPID  C  PRI  STIME  TTY  TIME CMD
  root 21557  1    0  60  Apr 12  -   0:00 db2wdog
  archive 21814 21557 0  60  Apr 12  -   0:30 db2sysc
  archive 22071 21814 0  60  Apr 12  -   0:00 db2dlock
  archive 24642 22328 0  60  Apr 12  -   0:04 db2loggr
  archive 31611 22071 0  60  Apr 19  -   0:05 db2gds
  root 22343  1    0    08:09 -   0:00 db2licd
  archive 34694 21601 0  10  13:13 -   1:09 db2agent (ARCHIVE)
  archive 34695 21601 0  10  13:13 -   1:09 db2agent (idle)
  archive 34949  1    0  10  13:13 -   1:20 db2agent (ARCHIVE)
  csr008 43938  1    0  10  13:08 -   3:40 /home/archive/sql1lib/bin/db2bp 44402 60
  csr001 43938  1    0  10  13:08 -   3:40 /home/archive/sql1lib/bin/db2bp 43679 60
  archive 22071 42414 0  60  Apr 12  -   0:00 db2ipccm
  archive 24642 22328 0  60  Apr 12  -   0:04 db2resyn
  archive 31665 42414 0  60  Apr 19  -   0:05 db2pfchr
  archive 31666 41414 0  60  Apr 19  -   0:05 db2pclrn
```

Displaying ADSM for AIX processes

The following processes are associated with ADSM for AIX:

- dsmserv** The ADSM for AIX server. Several occurrences of this process may be active.
- dsmlabel** The ADSM for AIX command used to label storage volumes. This process must not be active at the same time that the *dsmserv* process is active.

Refer to “Starting ADSM” on page 34 for instructions about how to restart ADSM for AIX.

Use the following command to display process status:


```
ps -ef | grep dsm
```

Figure 22. Monitoring ADSM for AIX Processes with the ps Command

The output of the command should be similar to the following example:

UID	PID	PPID	C	PRI	STIME	TTY	TIME	CMD
root	2880	24106	0	60	Apr 15	-	0:01	dsmserv quiet
root	11324	24106	0	60	Apr 15	-	0:00	dsmserv quiet
root	13626	24106	0	60	Apr 15	-	0:00	dsmserv quiet
root	14126	24106	0	60	Apr 15	-	0:00	dsmserv quiet
root	26413	24106	0	60	Apr 15	-	0:00	dsmserv quiet

Monitoring system load

The *uptime* command displays an estimate of the system load. Type the following command at the prompt:

```
uptime
```

Figure 23. uptime Command

Information reported by *uptime* includes the number of users on the system and load averages for the latest minute, 5 minutes, and 15 minutes.

```
11:40AM up 4:39, 2 users, load average: 0.17, 0.10, 0.09
```

Figure 24. uptime Command Output

In the example, the system load in the last minute is 17 percent, over the last 5 minutes is 10 percent, and over the last 15 minutes is 9 percent.

High load averages usually mean that the system is being used heavily and probably means users are experiencing slow response time. For more information about the *uptime* command, type **man uptime** at the prompt or refer to your AIX documentation.

Monitoring memory usage

To monitor memory use and paging activity, use the *vmstat* command along with the *ps* command. Type the following command at the prompt:

```
vmstat
```

Figure 25. vmstat Command

The most important pieces of information reported by *vmstat* are the number of processes running and the number of page-outs and swaps.

procs		memory		page				faults				cpu				
r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa
0	0	5143	26542	0	0	0	0	0	0	135	179	78	1	5	90	5

Figure 26. *vmstat* Command Output

After running *vmstat*, use *ps* to display the processes that are generating the memory usage displayed by *vmstat*.

For more information about the *vmstat* command, type **man vmstat** at the prompt or refer to your AIX documentation.

Monitoring system paging space

To monitor system paging space, use the *lsps* command. Type the following command at the prompt:

```
lsps -a
```

Figure 27. *lsps* Command

The *lsps* command reports the size of the page space and the percent used.

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
hd6	hdisk0	rootvg	64MB	17	yes	yes	lv

Figure 28. *lsps* Command Output

For more information about the *lsps* command, type **man lsps** at the prompt or refer to your AIX documentation.

Monitoring disks and file systems

The *df* command reports capacity numbers for disks and file systems, including cache file systems. Type the following command at the prompt:

```
df
```

Figure 29. *df* Command

The command shows the percentage each file system is used and how much space remains in each file system.

Filesystem	Total KB	free	%used	iused	%iused	Mounted on
/dev/hd4	24576	11112	54%	939	15%	/
/dev/hd9var	53248	49964	6%	153	1%	/var
/dev/hd2	548864	10000	98%	29610	21%	/usr
/dev/hd3	12288	9876	19%	46	1%	/tmp
/dev/lv07	4096	3004	26%	66	6%	/home
/dev/lv00	1880064	423532	77%	42	0%	/arsacif/acif1
/dev/lv01	1875968	484860	74%	111	0%	/arsacif/acif2
/dev/lv02	1019904	551420	45%	106	0%	/arsacif/acif3
/dev/lv03	1019904	960896	5%	20	0%	/arsdb
/dev/lv04	1019904	960896	5%	20	0%	/arsdb_primarylog
/dev/lv05	1019904	960896	5%	20	0%	/arsdb_archivelog
/dev/cache1	323584	313272	3%	16	0%	/arscache/cache1
/dev/cache2	323584	313272	3%	16	0%	/arscache/cache2
/dev/cache3	323584	313272	3%	16	0%	/arscache/cache3
/dev/cache4	323584	313272	3%	16	0%	/arscache/cache4
/dev/cache5	323584	313272	3%	16	0%	/arscache/cache5
/cdrom/aix	380928	49220	87%	-	-	/usr/lpp/info

Figure 30. *df* Command Output

Monitor the file systems that support the OnDemand database on a regular basis. Before the file systems become fully utilized, you must add storage volumes to the file systems and allocate segments of the database to the new storage volumes. Refer to your DB2 for AIX documentation for information about how to segment the database.

Monitor the file systems that support the document cache on a regular basis. Before the cache file systems become fully utilized, you must add storage volumes to the file systems or define additional file systems for the cache. Use the *smit* command to add volumes to a file system. Refer to the *Installation and Configuration Guide* for information about how to define additional cache file systems to OnDemand.

For more information about the *df* command, type **man df** at the prompt or refer to your AIX documentation.

Monitoring I/O activity

The *iostat* command reports input/output statistics for disks. *iostat* shows you the time physical disks are active in relation to their average transfer rate. You can use the information reported by *iostat* to better balance input/output load between physical disks. Type the following command at the prompt:

```
iostat
```

Figure 31. *iostat* Command

For a report interval and each physical disk on the system, *iostat* reports the percentage of time the disk was active (tm_act), the amount of data read or written to the drive (Kbps), transfers per second to the physical disk (tps), and the total amount of data read and written (Kb_read and Kb_wrtn). The following is an example of the *iostat* Disk Utilization report.

Disks:	% tm_act	Kbps	tps	mtps	Kb_read	Kb_wrtn
hdisk0	0.0	0.1	0.0		5993	9452
hdisk1	0.0	0.3	0.0		31193	6812
hdisk2	1.9	2.1	0.8		13153	232508
hdisk3	0.0	0.0	0.0		0	0
hdisk4	0.0	0.0	0.0		4293	4936

Figure 32. *iostat* Command Output

You can display continuous reports using the following format. In the example, *iostat* displays reports every 30 seconds for the logical disk *hdisk2*.

```
iostat -d hdisk2 30
```

Figure 33. *iostat* Continuous Command

For more information about the *iostat* command, type **man iostat** at the prompt or refer to your AIX documentation.

Monitoring storage volumes

You can use the following commands to display reports about storage volume utilization.

- Space report.

The space report provides general information about the storage volumes you defined, including the status, capacity, and utilization of each storage volume.

```
query volume format=detailed
```

Figure 34. *ADSM for AIX Query Volume Command*

- Database report.

The database report displays the volumes assigned to the database.

```
query db format=detailed
```

Figure 35. *ADSM for AIX Query Database Command*

When these reports show the *ADSM for AIX* server is approaching the capacity of the storage volumes, you must label and check new storage volumes into the library.

For more information about monitoring storage volumes, refer to the *ADSM for AIX Administrator's Guide*

Monitoring the network

The *netstat* command monitors TCP/IP network activity on a server. This command is useful for monitoring and gathering information, such as the number of clients that are connected to the server, whether the network connection is active or waiting, and how much data is currently being transferred between the server and clients.

```
netstat -f inet
```

Figure 36. *netstat* Command

The *netstat* command displays a report similar to the following example.

```
Active Internet connections
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp    0  15972 pikes.bpc.boulde.1445  csr0018.bpc.boulde.1039 ESTABLISHED
tcp    0  4039  pikes.bpc.boulde.1445  csr0013.bpc.boulde.1049 ESTABLISHED
tcp    0    0  pikes.bpc.boulde.1501  elric.bpc.boulde.1635  ESTABLISHED
tcp    0    0  pikes.bpc.boulde.1445  csr0001.bpc.boulde.1026 ESTABLISHED
tcp    0    0  pikes.bpc.boulde.1026  pikes.bpc.boulde.4242  ESTABLISHED
tcp    0    0  pikes.bpc.boulde.4242  *.*                     LISTEN
```

Figure 37. *netstat* Command Output

The Local Address field contains the TCP/IP port address for the server. Clients connected to the server can be identified using the port address. In this example, the TCP/IP port address is 1445. Check the */etc/services* file to verify the port address.

For more information about the *netstat* command, type **man netstat** at the prompt or refer to your AIX documentation.

Chapter 8. Performance and tuning

This section describes the commands that you can use to collect information about OnDemand server operation and resources, to help optimize the performance of OnDemand.

Good client and server performance depend on how well you balance CPU, I/O, and memory resources so that all processes obtain the share of these resources they need. Good performance also depends on you recognizing the resource that is the culprit when performance is degraded. For example, you might conclude that poor system response results from contention for the CPU when in fact, the CPU is waiting on I/O. Alleviating the I/O bottleneck results in better throughput.

There are four major subsystems interacting on an OnDemand server that will be the subject of most performance and tuning activities.

- The operating system kernel (AIX/6000). Primarily responsible for managing processes. The two resources that most affect operating system response are CPU and memory.
- The database management system (DB2 for AIX). A set of processes responsible for the OnDemand database and the document cache. DASD I/O is the resource that most affects the performance of DB2 for AIX.
- The storage management system (ADSM for AIX). A set of processes responsible for optical storage volumes. CPU and DASD I/O are the resources that most affect the performance of ADSM for AIX.
- The network (TCP/IP). High transfer rates and otherwise overloaded networks reflect badly on client/server applications. Network I/O is the resource that most affects the performance of TCP/IP.

Optimizing the performance of the operating system, databases, networks, client software, and server programs may seem complex and can become time consuming. The following tools and recommended documentation can help ease the task of collecting and evaluating performance information. Remember, *balancing* resources is the key to good performance. You have a finite amount of resources to allocate among processes. Not every process can run at the highest priority. There is only so much disk and network I/O bandwidth the system can provide to processes. Your challenge is to make sure the processes that need priority and bandwidth get it when they need it, in the amount and for the duration required.

AIX/6000 tuning considerations

Base Operating System performance package

AIX/6000 provides a set of performance tools which may be used to evaluate and tune AIX system performance. For more detailed information about each of the tools, type **man *toolname*** at the prompt. Replace the string *toolname* with one of the following tools.

filemon

Monitors a trace of file system and I/O system events, and reports performance statistics for files, virtual memory segments, logical volumes, and physical volumes. The *filemon* program is useful to determine why applications are I/O bound.

fileplace

Displays the placement of a file's blocks within a file system. Logically contiguous files may be physically fragmented on disk, depending on the available free space at the time the file is created. The *fileplace* program can be used to examine and assess the efficiency of a file's placement on disk.

lvedit

Provides precise control over the creation and placement of logical volumes on physical volumes. As an interactive tool, The *lvedit* command allows users to view current logical volume placements, make tentative changes to the placements (creating, removing, or extending logical volumes), and commit those changes.

netpmon

Monitors a trace of system events, and reports on CPU usage, network device-driver I/O, internet socket calls, and NFS I/O. In its normal mode, The *netpmon* command runs in the background while one or more application programs or system commands are being executed and monitored. When tracing is stopped via a *trcstop* command, *netpmon* generates all specified reports.

rmss

Provides a means to simulate an RISC System/6000 with various size memories, without having to extract and replace memory boards. The *rmss* program provides a facility to simulate running an application over a range of memory sizes, displaying, for each memory size, performance statistics, such as the response time of the application and the amount of paging. The *rmss* program is designed to help you determine how much memory a RISC System/6000 needs to run a given application with an acceptable level of performance.

svmon

A virtual memory monitor. When invoked, *svmon* captures a snapshot of the current contents of both real and virtual memory, and creates a summary report. The *svmon* program can help identify the processes or segments that consume the most real memory.

AIX Performance Toolbox/6000

The AIX Performance Toolbox/6000 (PTX/6000) product provides a graphical interface for monitoring AIX systems and networks. PTX/6000 includes real time performance monitoring tools and performance tuning controls. For more information about PTX/6000 (product number 5696-623), refer to the *AIX Performance Toolbox/6000: User's Guide*

AIX Performance Monitoring and Tuning Guide

For more information about performance monitoring and tuning the AIX environment, refer to the *AIX Performance Monitoring and Tuning Guide*. The Guide provides concepts and techniques involved in assessing and tuning the performance of AIX systems.

DB2/6000 tuning considerations

The following information introduces statistical information you can collect and some of the parameters that can affect the performance of DB2 for AIX. For detailed information, refer to the *DB2 for AIX Administration Guide*.

Database system monitor

DB2 for AIX provides a set of database system monitor APIs for gathering statistics about the operation of databases. Information can be gathered to assess the status and improve the performance of DB2 for AIX, databases, tables, and applications.

The database system monitor can be controlled and accessed by using commands provided with the DB2 for AIX command line processor interface. The following are examples of commands you can use to collect information about DB2 for AIX.

- To display the parameters and values for the OnDemand database and list the configuration parameters that you can change:

```
GET DATABASE CONFIGURATION FOR ARCHIVE
```

Figure 38. Get Database Configuration Command

- To display the parameters and values for DB2 for AIX and list the configuration parameters that you can change:

```
GET DATABASE MANAGER CONFIGURATION
```

Figure 39. Get Database Manager Command

- To list statistics about DB2 for AIX at a specific point in time, reporting on databases, applications, tables, and DB2 for AIX:

```
GET SNAPSHOT
```

Figure 40. Get Snapshot Command

For more information about database commands, refer to the *DB2 for AIX Command Reference*. For more information about understanding and tuning database parameters, refer to the *DB2 for AIX Administration Guide*.

Configuration parameters

The default values of the parameters provided with DB2 for AIX are oriented toward machines with small memory (for example, 64 megabytes) and dedicated as database servers. You may need to modify the values to support an environment with large databases, provide high performance for a specific application, and if your hardware configuration is different from the default configuration. For more information about changing configuration parameters, see the *DB2 for AIX Administration Guide*

Index statistics

DB2 for AIX has utilities to improve the way data are physically stored on disks and to update the statistical information used by the DB2 for AIX query optimizer. OnDemand automatically runs these utilities as part of database loading.

REORG utility

This utility rearranges data into a physical sequence according to a specific index. This can allow queries to be performed more efficiently. When you run the REORG utility, OnDemand uses the index information you specified when you created the OnDemand database to optimize the records in the database.

RUNSTATS utility

This utility carries out several tasks to optimize access to data, using statistical information maintained by DB2 for AIX.

ADSM for AIX tuning considerations

The number of parameters that affect performance and may be set within ADSM for AIX are quite small. Tuning AIX/6000 and TCP/IP options to get good performance with ADSM for AIX can become complex, but the payback can be great.

Please refer to the *ADSM Performance Tuning Guide* for detailed information about tuning ADSM for AIX on the RS/6000 with TCP/IP. It provides details about the various parameters affecting performance and defaults and recommended values for good performance.

TCP/IP tuning considerations

OnDemand server

Tuning client and server network performance can become complex, especially considering the different types of hardware and software that an OnDemand system can support.

Please refer to the documentation provided with the TCP/IP software and the network hardware for specify OnDemand client and server operating environments, to obtain details about monitoring and tuning network-related parameters. For example:

- The *AIX/6000 System Management Guide: Communications and Networks* publication provides information about configuring TCP/IP on an OnDemand server
- The *ADSM for AIX Performance Tuning Guide* provides details about tuning ADSM for AIX for TCP/IP.
- The *IBM TCP/IP Version 2.1 for DOS: Installation and Administration Guide* contains information that can help you optimize the network performance for personal computers running the OnDemand Windows client program.

Chapter 9. Backup and recovery

Overview

This chapter describes backup and recovery for OnDemand and provides recommendations about methods and procedures that the system administrator can use to ensure the following critical OnDemand components can be recovered, if needed:

- OnDemand software.
- OnDemand control files, created or modified during installation, configuration, and ongoing operation of OnDemand.
- The OnDemand database.
- The ADSM database.
- Cache file systems.
- Report files and resources.

The *Installation and Configuration Guide* contains information about the file systems and directories that support the components of OnDemand.

The *Installation and Configuration Guide* contains information about how to set up the cron facility to automatically start jobs that create backup images of the databases on a regular schedule.

OnDemand server software

If a media failure or some other unforeseen event occurs, you may be required to restore OnDemand software programs, unmodified OnDemand shell scripts, and ADSM for AIX and DB2 for AIX software. You can use the original product tapes to accomplish this task.

It is important that you store the original product tapes in a safe location. We recommend that you register OnDemand as part of your business recovery plan and store the original product tapes with other vital information systems records.

OnDemand control files

When you installed and configured OnDemand, you modified several files to customize OnDemand to operate in your environment. We recommend that you make a backup copy of all modified files immediately after you have verified the installation of OnDemand. Please see OnDemand Configuration and Shell Script Files in the *Installation and Configuration Guide* for a list of files that you might update when you configure an OnDemand system.

In addition, if you periodically make changes to or create new AIX/6000, OnDemand, ADSM for AIX, and DB2 for AIX control files, you may find it helpful to backup these files on a regular basis, perhaps once a week. Directory trees that may contain

modified files include `/usr/lpp/ars`, `/usr/lpp/adsm`, `/usr/lpp/admserv`, and `/home/archive`. Your AIX system administrator probably schedules regular backups of AIX file systems using the `mksysb` command. Check with the system administrator to make sure that AIX/6000, DB2 for AIX, ADSM for AIX, and OnDemand control files required for OnDemand operations are included in this backup.

You can use several different AIX commands along with the cron facility to make backup copies of files and schedule regular backups of selected file systems, directories, and files. For example, the following command can be used to create a backup copy of files in the `/usr/lpp/ars` file directory tree:

```
tar -cv -f/dev/rmt0 /usr/lpp/ars
```

Figure 41. `tar` Command

In the example, the `tar` command copies all files under the directory tree to tape. Error messages are written to the console. The *AIX/6000 System Management Guide: Operating System and Devices* publication provides details about backup and restore concepts and commands.

The OnDemand database

DB2 for AIX table spaces

DB2 for AIX table space support provides enhanced flexibility and improved performance for your application group data. For example, after you store a report in OnDemand, you can create a backup image of the table that changed during the load process, rather than creating a backup image of the entire database. You can also create an incremental backup image of the database, which contains only those tables that changed since the last backup image. Because the backup image only contains the changes made to the database, the backup process typically runs much faster than a full backup.

OnDemand creates one table space for each segment of application group data. After OnDemand closes the segment and you back up the table space, you do not need to back up the table space again, unless it is recovered or restored.

When you use the incremental table space backup capability, we recommend that you backup the OnDemand database after each report file load. If your schedule does not permit you to run the backup command after each report file load, we recommend that you backup the database once a day (assuming that you load multiple reports each day). While incremental backup images can be used to recover the database, we recommend that you periodically create a full backup image of the database. A full backup image of the database is the quickest way to recover the database, in the event that you need to do so. However, if your OnDemand database is very large and cannot be backed up in a reasonable amount of time or requires many storage volumes to back up, you may find that maintaining full backup images of the database is not possible.

The *Installation and Configuration Guide* provides details about how to configure the system to support table spaces.

Database backup

OnDemand provides the `ars_db` command so that you can create backup images of the OnDemand database. The `ars_db` command supports table space and full database backups:

- OnDemand provides support for incremental table space backups and full database backups.
- An online backup can be taken when other applications or processes are connected to the database. That is, other applications and processes can continue to read or modify data while the backup is in process. An online backup can be scheduled with the cron facility.
- During an offline backup, only the backup task is connected to the database. Before starting an offline backup, we recommend that you stop the OnDemand system to make sure that no other applications or processes are connected to the database.
- When you back up the database with the `ars_db` command, OnDemand removes the log files from the archived log file directory, releasing the space taken by files that are no longer needed. However, if you use ADSM to manage DB2 log files, the policy domain determines when archived log files are removed.

If your production schedule allows, we strongly encourage you to create offline backups on a regular schedule, perhaps once a week. Taking weekly offline backups can reduce the time required to rebuild table spaces or the database, if you need to do so. We recommend that you write offline backup images to removable media or ADSM-managed storage. Keep backup images in a safe place until the next time that you create an offline backup image of the table space or database.

If your schedule does not provide time to take offline backups (that is, your system must always remain available to users), you should take online backups on a regular schedule. The *Installation and Configuration Guide* shows how to use the cron facility to create online backups of the database with the `ars_db` command, automatically on a regular schedule.

Chapter 17, “`ars_db` command reference” on page 103 provides details about the `ars_db` command, parameters, and options.

The *DB2 for AIX Administration Guide* provides details about backing up a database.

Using ADSM to manage backup images

You can use ADSM to manage DB2 for AIX backup image files. This eliminates the need for you to manage DB2 for AIX backup image files on disk. When you use the `ars_db` command to create table space backup images or backup the database, you can specify that you want ADSM to manage the database or table space backup images.

Before you can use ADSM to manage DB2 for AIX backup image files, you must define an ADSM storage hierarchy to manage the files. The storage hierarchy includes definitions that identify the type of media and storage devices that ADSM stores files on, the length of time that ADSM maintains files, and the number of backup copies that ADSM maintains. Before you schedule a backup command, you must make sure that ADSM storage volumes exist with sufficient free space to hold the backup image.

The *Installation and Configuration Guide* provides details about how to configure the system to use ADSM to manage backup image files. Refer to Chapter 17, “ars_db command reference” on page 103 for details about how to use the ars_db command to backup table spaces to ADSM-managed storage.

Example: creating a full offline backup of the database

You can use the ars_db command to create a full offline backup image of the OnDemand database. If you are managing DB2 log files on disk, after completing the backup command, OnDemand removes the log files from the archived log file directory.

An offline backup of the database must be run at a time when no users are accessing the system and there are no other applications connected to the database. We recommend that you stop the OnDemand system before starting the offline backup.

The following example show how to use the ars_db command to create a full offline backup of the OnDemand database on a tape in the device */dev/rmt0*.

```
/usr/lpp/ars/bin/ars_db -v -y /dev/rmt0
```

Figure 42. Offline Database Backup Command

You must place a blank, formatted tape storage volume in the tape device before you start the ars_db command. The ars_db command writes processing messages to the console. After the offline backup completes, you can restart the OnDemand system. Chapter 17, “ars_db command reference” on page 103 provides details about the ars_db command, parameters, and options. The *DB2 for AIX Administration Guide* provides details about backing up a database.

Database logging

DB2 for AIX uses transaction logging to record changes to the OnDemand database. The information in the log file is used to recover from corruption of data in the database. Logging ensures that no data is lost. By combining the information in the log files with a backup copy of the database, the OnDemand database can be recovered to any point in time.

The OnDemand database and the DB2 for AIX log files should reside on different physical volumes. The database backup image should be written to removable media. Unless multiple disk and tape volumes are damaged or lost **at the same time**, there is no possibility of losing the information contained in the OnDemand database.

The *DB2 for AIX Administration Guide* provides details about database logs.

Using ADSM to manage archived log files

You can use ADSM to manage DB2 for AIX archived log files. This eliminates the need for you to manage the log files on disk.

Before you can use ADSM to manage DB2 for AIX log files, you must define an ADSM storage hierarchy to manage the files. The storage hierarchy includes definitions that identify the type of media and storage devices that ADSM stores files on, the length of time that ADSM maintains files, and the number of backup copies that ADSM maintains. Before DB2 for AIX creates archived log files, you must make sure that ADSM storage volumes exist with sufficient free space to hold the files.

The *Installation and Configuration Guide* provides details about how to configure the system to use ADSM to manage DB2 log files.

Database recovery

This topic provides an overview of the different recovery methods that can be used in the event there is a problem involving the database.

Note

We recommend that you contact OnDemand technical support to discuss the strategies available to you when there are problems with the database and help implement a backup and recovery plan that is best suited to your business and operating environment. The *DB2 for AIX Administration Guide* provides details about database backup and recovery.

Typically you will need to recover the OnDemand database because of media and storage problems, power interruptions, and application failures. When a problem occurs that damages or corrupts the database in some way, you must rebuild the database. The rebuilding of the database is called recovery. There are two types of database recovery:

- The first type recovers from failures that occur while update transactions are taking place. For example, a system failure occurs while update transactions are taking place. The database is left in an inconsistent and unusable state and must be moved to a consistent and usable state before you can permit users to access the system.

The transaction logs help correct this type of failure by allowing the transactions received before the failure to either be reapplied to the database or to be *rolled-out*. Rolling-out transactions is a way to return the database to the state it was in before the transaction that caused the failure.

This type of recovery is done with the DB2 for AIX `RESTART DATABASE` command. If you want this type of recovery to occur in every case of a failure, you can use the automatic restart enable database configuration parameter. The default for this configuration parameter is that the `RESTART DATABASE` routine will be started every time it is needed. Once enabled, you do not need to do anything to have this command done at the time of a failure.

- The second type of recovery deals with corruption of the OnDemand database and is usually caused by media failure. For example, one of the magnetic storage volumes that belongs to the database volume group becomes damaged and unusable. To recover from this type of failure, an administrator must intervene to recover the database.

The combination of the DB2 for AIX log files and a full backup copy of the database can be used to recreate the OnDemand database to any particular point in time. First, the latest full backup image of the database rebuilds the database to a point in time. Then, a roll-forward recovery restores all of the units of work that occurred since the backup image was created. This allows you to restore the database to a state identical to the time of the failure.

The OnDemand database and the DB2 for AIX log files should reside on different physical volumes. The database backup image should be written to removable media. Unless multiple disk and tape volumes are damaged or lost **at the same time**, there is no possibility of losing the information contained in the OnDemand database.

Factors affecting recovery

To decide which database recovery method to use, you must consider the following key factors:

- How near to the time of failure you will need to recover the database?

When you restore a full backup copy of the database, the database is only as current as the time that the last backup was made.

To restore the database to the time of a failure, you must use the log files to reapply changes that were made to the database since the backup copy was created. You can reapply the changes to the end of the logs or to a point in time. A point in time recovery may be useful if an application corrupts the database and you do not want to reapply its changes.

- How much time is spent associated with recovery?

Your recovery plan should allow for regularly scheduled backups, since backing up the database requires time and system resource.

You can do a backup while the database is either online or offline. If it is online, users can access the system and other processes can connect to the database and read and modify data while the backup task is running. If the backup is performed offline, only the backup task can be connected to the database. Users cannot access the system and other processes cannot connect to the database while the offline backup task is running.

- How much storage space can be allocated for backup copies and archived log files?

To restore the database, you must allocate enough free disk space to hold the backup copy of the database and the restored database. To roll-forward transactions requires space to hold the backup copy of the database, the restored database, and all of the archived log files created between backup copies of the database.

- Table space level or full database level backup.

With a table space backup, you can specify one or more table spaces to be backed up, rather than the entire database. You can then restore selected table spaces to a state identical to the time the backup was made. However, those table spaces not selected at the time of the backup will not be in the same state as those that were restored.

We strongly encourage you to speak with OnDemand technical support, before you implement a backup and recovery plan that includes table space backup and recovery.

The *DB2 for AIX Administration Guide* provides details about recovering a database.

The ADSM for AIX database

ADSM for AIX maintains a database that contains information about the devices and files it manages. After you load a report file into an application group that requires archive media support, ADSM for AIX updates the database and stores a copy of the file on a storage volume. When you define devices and register nodes, ADSM for AIX updates the database. When ADSM for AIX maintains storage volumes, it updates the database with status information about files and storage volumes. The ADSM database is critical to proper operation of ADSM for AIX in storing objects on and retrieving objects from archive media.

Mirroring the ADSM database is strongly encouraged. When you mirror the database, ADSM for AIX replicates the database onto different physical storage. ADSM for AIX automatically keeps track of and refreshes both copies of the database. When you configure physical storage so that ADSM for AIX can mirror the database on different physical devices and adapters, you can provide protection for the database due to failure of a single device. With mirroring, ADSM for AIX can continue operation without interruption if a database volume fails by using a mirrored copy of the failed volume. Mirroring requires additional storage space for the mirrored volumes. The *ADSM for AIX Administrator's Guide* provides details about mirroring the ADSM database.

To protect the information in the database, and ensure that it can be restored if a disaster occurs, you must periodically create a backup copy of the database. You can recover the database to its most current state or to a specific point in time with the backup copy.

- A full backup image of the database should be taken after you install and configure ADSM for AIX with OnDemand. In addition, we recommend that you create a full backup of the database on a regular schedule, such as once a week. A full backup copy of the database should be written to removable media.
- An incremental backup image of the database records changes that occurred since the last backup of the database (full or incremental). You can create a maximum of 32 incremental backups between full backups of the database. If you write incremental backup images of the database to disk, make sure that the disk is on a different controller than any of the database or recovery log volumes.

There are several factors to consider when you decide the type and frequency of backups.

- A full backup takes longer to run than an incremental backup.
- Recovery time is faster with a full backup. Incremental backups increase the time it takes to recover the database because a full backup must be loaded first, followed by some or all of the incremental backups.
- A full backup is required under specific conditions. You must create a full backup after installing and customizing ADSM for AIX with OnDemand. You can run up to 32 incremental backups between each full backup.

We recommend that you back up the database after you load report files into OnDemand and following ADSM for AIX maintenance of storage volumes (expiration and reclamation). Under typical conditions, you should back up the database each day.

ADSM for AIX includes a central scheduling component that allows the automatic processing of administrative commands, such as database backup. Each administrative command is called an event. Each scheduled event is tracked by the server and recorded in the database. You set up an administrative command schedule by defining schedule parameters, such as the start day, date, and time, specifying the command to be executed, such as the BACKUP DATABASE command, and activating the schedule. The *ADSM for AIX Administrator's Guide* provides details about scheduling operations.

OnDemand provides the `ars_adsm` command to create a full backup of the database. You can submit the command using the AIX `at` command or run the command automatically on a regular schedule using the AIX cron facility. If you invoke the data loading program from a shell script, you can add a step to the script that runs the command after loading report files into OnDemand. The *Installation and Configuration Guide* describes how to set up the cron facility to start the `ars_adsm` command and create a full backup image of the database automatically on a regular schedule. The *Administrator's Reference* provides details about the `ars_adsm` command, parameters, and options.

ADSM recovery log

The recovery log is critical to the operation of the ADSM for AIX server. If the recovery log is unusable, the entire server is unavailable. With the recovery log available, and a restored backup image of the database, you can recover the database to its most current state.

To ensure fast recovery time and high availability of the database, you should always mirror the recovery log. Mirroring the recovery log requires much less space than mirroring the database. If you do not mirror the recovery log, you should allocate the recovery log on a disk other than the one on which the database resides. The *ADSM for AIX Administrator's Guide* provides details about mirroring the ADSM recovery log.

When a database backup is completed, recovery log records preceding the backup are deleted, freeing up recovery log storage for reuse. Taking frequent database backups

reduces recovery log storage requirements, and reduces the time required to recover the database.

ADSM volume history file

Volume history information is vital for recovery of a lost or damaged database. The volume history file contains information that ADSM for AIX needs about the volumes to use for database backups. The volume history file contains information that you need to know about which storage pool volumes to audit after a recovery.

ADSM for AIX cannot obtain volume history information from the database during a restore of the database. Therefore, you should store at least one backup copy of the volume history file, on a disk other than the one on which the database resides.

The *ADSM for AIX Administrator's Guide* provides details about creating volume history backup files.

ADSM device configuration file

When you define, update, or delete a device class, drive, or library, ADSM for AIX updates the database and makes an entry in the device configuration file. To restore the database, ADSM requires a definition for the device class from which backup data is to be read. This definition is in the device configuration file.

When the database is being restored, no definitions can be read from the database. Therefore, you should have at least one backup copy of the device configuration information, on a disk other than the one on which the database resides.

The *ADSM for AIX Administrator's Guide* provides details about creating device configuration backup files.

ADSM database recovery

Recovering using mirrored copies of the database

If a database volume fails because of media failure and you have enabled mirroring, you can recover the database by using mirrored copies of the database. After fixing the failing device, you can allocate space for the new mirrored copy and define the volume to ADSM for AIX. After you define the volume to ADSM for AIX, the server synchronizes the volume with the database.

Recovering using backup copies of the database

ADSM for AIX provides commands to recover the database, should a catastrophic failure occur. These commands restore the database from the latest available full backup copy, apply all incremental backups that apply, and use the recovery log to apply any changes made to the database since the last backup was created.

If you restore the database to its most current state, ADSM for AIX automatically synchronizes the database and storage volumes.

If you restore the database to a specific point in time, you must audit all storage pool volumes to check for and resolve any inconsistencies between database information and storage pool volumes. Depending on the number of storage pool volumes and the amount of activity that occurred after the database backup that you restored, the audit may require a significant amount of time.

To perform a database recovery, you should have the following information, preferably stored offsite:

- Back up volumes of the database
- Copy storage pool volumes
- Server options file
- Volume history file
- Device configuration file
- Output from ADSM for AIX commands that provide details of the database and recovery log setup

The *ADSM for AIX Administrator's Guide* provides details about recovering the ADSM database.

Archived report files

OnDemand can store copies of reports and resources in the document cache and on archive media.

- The primary purpose of the document cache is short-term, high-speed storage and retrieval. The document cache consists of magnetic storage volumes maintained by OnDemand on one or more object servers.
- The primary purpose of archive media is long-term storage and retrieval. Reports on archive media can also be used as backup copies, in the event cache storage becomes corrupted or unavailable. Archive media consists of optical or tape storage volumes managed by ADSM.

Most customers copy reports to cache storage and archive media at the same time, when they load a report into the system.

OnDemand can retrieve a copy of a report from archive media after the report has been deleted from cache storage or if the copy on cache storage is unavailable. However, you must properly configure your system to support multiple copies of reports. You must install and configure ADSM, add archive media devices to your system, and configure OnDemand to use archive media. You configure OnDemand to use archive media by defining storage nodes in ADSM, assigning application groups to the appropriate storage sets, and setting application group data migration and caching information.

Note: If you do not copy reports to archive media, we recommend that you use ADSM to protect the cache file systems. ADSM can provide a range of storage management services. You can use ADSM to make backup copies of cache file

systems. If a cache storage device fails, you can use ADSM to restore all or part of the cache file system. However, without a copy of reports on archive media, users cannot retrieve reports until cache storage is restored.

The document cache

The document cache is the primary, short-term storage location for reports.

If you do not copy reports to archive media when you store them in OnDemand, you need to consider how you can recover the reports in the event you need to do so (for example, if a cache storage device fails).

The document cache can be protected by maintaining it on RAID storage subsystems. RAID storage can provide excellent availability, allowing users to access reports even if a disk or controller fails. The *Introduction and Planning Guide* provides information about using RAID storage with OnDemand. However, RAID storage is not fail safe. There may be situations when, because of multiple disk or controller failures, users cannot access reports. We encourage you to use ADSM to maintain a copy of reports on archive media.

Reports on archive media

The primary storage node identifies the object server and ADSM node where OnDemand stores the primary copy of a report on archive media. OnDemand retrieves the primary copy of the report from archive media after the report has been deleted from cache storage. Customers with critical, high availability requirements may need to create a backup or secondary copy of the report on archive media. The backup or secondary copy can be used if the primary copy becomes corrupted or unavailable.

There are two methods that you can use to create a backup or secondary copy of reports:

- Define a *copy storage pool* to ADSM for AIX. With this method, ADSM for AIX manages a backup copy of files that are stored in a primary storage pool independently and transparently to OnDemand. The backup copy is stored in a copy storage pool that can be used to restore the original files if they become damaged, lost, or unusable. The copy storage pool can be assigned to the same library as the primary storage pool. However, you would typically assign the copy storage pool to a different library. You can copy data from one or more primary storage pools to the same copy storage pool. Copy storage pools require additional space in the ADSM for AIX database. A copy storage pool must reside on the object server where the primary storage pool resides.

ADSM for AIX includes a central scheduling component that allows the automatic processing of administrative commands, such as copying data from a primary storage pool to a copy storage pool. Each administrative command is called an event. Each scheduled event is tracked by the server and recorded in the database. You set up an administrative command schedule by defining schedule parameters, such as the start day, date, and time, specifying the command to be executed, such as the BACKUP STGPOOL command, and activating the schedule.

| The *ADSM for AIX Administrator's Guide* provides details about defining and
| managing a copy storage pool, and storage pool backup and recovery, and
| scheduling operations.

- Assign the primary storage node to a secondary storage node. With this method, ADSM maintains a secondary copy of files that are stored in a primary storage node. The secondary copy can be used if the primary copy becomes damaged, lost, or unusable. However, administrator intervention is required before OnDemand can use the secondary copy. For this reason, we discourage the use of secondary storage nodes.

Chapter 10. Adding storage volumes to a library

During installation, you prepared storage volumes for ADSM for AIX. Over time, as you archive report files in OnDemand, ADSM for AIX uses the initial set of storage volumes that you prepared. You must then add storage volumes to the library so that OnDemand can continue to archive files. Preparing new storage volumes for ADSM for AIX involves labeling the storage volumes and registering them with ADSM for AIX. Before you add storage volumes to a library, we recommend that you stop the OnDemand system.

The topics included in this section provide general instructions about how to label optical storage volumes and check them into a library. Please refer to the *ADSM for AIX Administrator's Guide* for details about managing storage devices, libraries, and storage volumes.

Note

If there are no slots available to place new storage volumes in a library, you must either add a library or remove storage volumes that are not likely to be accessed from an existing library. The *Installation and Configuration Guide* provides instructions about how to configure a new library on the RS/6000 and define the library to ADSM for AIX. Refer to Chapter 11, "Removing storage volumes from a library" on page 75 for instructions about how to remove storage volumes from a library.

Stop OnDemand

Before you begin, follow the instructions in "Stopping OnDemand" on page 36 to stop all of the processes related to OnDemand and remove mounted storage volumes from a library.

Label storage volumes

After stopping OnDemand, ADSM for AIX, and removing mounted storage volumes from the library, complete the following steps to label new storage volumes.

1. Login to AIX as the *root* user.
2. Change to the */usr/lpp/adsmserve/bin* directory.
3. Run the *dsmlabel* program to label storage volumes. The following example shows how to label optical storage volumes in an IBM 3995-C62 optical library:

```
dsmlabel -library=/dev/1b0
         -drive=/dev/rop0,1 -drive=/dev/rop1,2
         -keep
```

Figure 43. Labeling Storage Volumes

Replace the string `1b0` with the device name of the library. Replace the strings `op0` and `op1` with the element addresses of the optical drives. The `-keep` option causes the program to keep volumes inside the library after they are labeled.

4. The `dsmlabel` command prompts you to place a blank optical platter in each drive. Any platter with a pre-existing label is rejected and the next unlabeled platter is requested.
5. The `dsmlabel` command prompts you to type a label for the storage volume. All optical storage volumes must have unique labels. Labels can be from 1 to 32 characters in length. You can use the characters A through Z and 0 through 9. The label name cannot include embedded blanks. ADSM for AIX converts lowercase letters in a volume label to uppercase. We recommend that you maintain a central log with information about storage volumes. The log should contain information such as the date the volume was labeled, the type of media, the capacity, the library, the volume label, and the type of data to be written on the storage volume.
6. Repeat step 4 to label additional storage volumes.
7. Press **Enter** to exit the `dsmlabel` program when you have finished labeling storage volumes.

Refer to the *ADSM for AIX Administrator's Reference* for details about the `dsmlabel` command.

Check storage volumes into the library

Issue the `CHECKIN LIBVOL` command to make storage volumes available to ADSM for AIX. This command causes ADSM for AIX to check the storage volumes into the library and record the identifiers and status of the volumes in the ADSM database.

To check storage volumes into the library, you must start ADSM for AIX and the administrative client. We recommend that you start an administrative client in console mode and another administrative client in interactive mode. You can review messages generated by the maintenance tasks in the console window. You can enter ADSM for AIX commands in the interactive window.

1. Login to AIX as the `root` user.
2. Start ADSM for AIX, if it is not already started. For example:

```
/usr/lpp/ars/bin/ars_adsm -sv
```

3. Start an ADSM for AIX administrative client in console mode. Enter the following command at the prompt of an `aixterm` window:

```
dsmadm -consolemode
```

4. Enter an ADSM for AIX administrator userid and password when prompted.
5. Start an ADSM for AIX administrative client in interactive mode. Enter the following command at the prompt of an aixterm window:

```
dsmadm
```

6. Enter an ADSM for AIX administrator userid and password when prompted.
7. Issue the CHECKIN LIBVOL command to cause ADSM to scan a library for new storage volumes. For example:

```
checkin libvol archlib0 status=scratch search=yes
```

Figure 44. Check In Library Volume Command

Enter the command at the ADSM for AIX administrative client *adsm>* prompt. Replace the string *archlib0* with the name of the library.

8. ADSM for AIX processes all of the storage volumes in the library. ADSM for AIX displays status information in the console window, including the label of each volume checked into the library.
9. After ADSM for AIX completes scanning the library and registering all of the new storage volumes in the database, you can issue the QUERY LIBVOL command to list the volumes in the library and status information about the storage volumes.
10. Close the ADSM for AIX console and administrative client. Enter **QUIT** at the *adsm>* prompt to stop the client. Press the **Ctrl** and **C** keys to close the console.

Restart OnDemand

When you have completed labeling and checking storage volumes into the library, you can restart OnDemand. Follow the instructions in “Starting OnDemand” on page 33 to restart all of the processes related to OnDemand.

Chapter 11. Removing storage volumes from a library

The following topics describe how to remove storage volumes from an optical library. You typically perform these steps when all of the storage volumes in the library are fully used. You must remove at least one storage volume from the library, before you can place a new, scratch storage volume in the library.

Note

If you determine that no storage volumes can be removed from the library, you must add a library so that ADSM for AIX can continue to maintain data for OnDemand. The *Installation and Configuration Guide* provides instructions about how to configure a new library on the RS/6000 and define the library to ADSM for AIX.

Select the storage volumes to remove

Use the ADSM for AIX QUERY VOL command to determine the status of the storage volumes in a library. First, run the command to list all the storage volumes in the library. Then run the command to list information about selected storage volumes. The command reports the date a selected storage volume was last written to and accessed. When you need to remove full volumes to make room for new storage volumes, you generally remove volumes with the oldest access date.

To select volumes, start ADSM for AIX and the administrative client. We recommend that you start an administrative client in console mode and another administrative client in interactive mode. You can review messages generated by the maintenance tasks in the console window. You can enter ADSM for AIX commands in the interactive window.

1. Log on to AIX as the *root* user.
2. Start ADSM for AIX, if it is not already started. For example:

```
/usr/lpp/ars/bin/ars_adsm -sv
```

3. Start an ADSM for AIX administrative client in console mode. Enter the following command at the prompt of an aixterm window:

```
dsmdmc -consolemode
```

4. Enter an ADSM for AIX administrator userid and password when prompted.
5. Start an ADSM for AIX administrative client in interactive mode. Enter the following command at the prompt of an aixterm window:

```
dsmadm
```

- 6. Enter an ADSM for AIX administrator userid and password when prompted.
- 7. Issue the ADSM for AIX QUERY VOL command to list the storage volumes in the library.

```
query vol stgpool=0DAIX1
```

Figure 45. Query Volume Command

Enter the command at the ADSM for AIX administrative client *adsm*> prompt. Replace the string 0DAIX1 with the name of the storage pool that contains the storage volumes for the library.

ADSM for AIX displays a report similar to the following:

Volume Name	Storage Pool	Device Class Name	Estimated Capacity (MB)	%Util	Volume Status
AVOL1	0DAIX1	ODLIB0	1,309.0	99.9	On-line
AVOL2	0DAIX1	ODLIB0	1,309.0	99.9	On-line
AVOL3	0DAIX1	ODLIB0	1,309.0	99.9	On-line
AVOL4	0DAIX1	ODLIB0	1,309.0	99.9	On-line
AVOL5	0DAIX1	ODLIB0	1,309.0	99.9	On-line
:					
AVOL38	0DAIX1	ODLIB0	1,309.0	99.9	On-line
AVOL39	0DAIX1	ODLIB0	1,309.0	24.3	Filling
AVOL40	0DAIX1	ODLIB0	1,309.0	5.5	Filling

Figure 46. Query Volume Command Output

- 8. Display information about specific storage volumes. For example:

```
query volume avo11 format=detail
```

Figure 47. Query Volume Command Detail

Enter the command at the ADSM for AIX administrative client *adsm*> prompt. Replace the string avo11 with the name of the storage volume you want additional details about.

ADSM for AIX displays a report similar to the following:

```

          Volume Name: AVOL1
          Storage Pool Name: 7YR
          Device Class Name: ODLIB0
          Estimated Capacity (MB): 1,309.0
              %Util: 99.9
          Volume Status: On-line
              Access: Read
          Pct. Reclaimable Space: 2.3
              Scratch Volume?: No
              In Error State?: No
          Number of Writable Sides : 2
          Number of Times Mounted : 48
              Write Pass Number: 2
          Approx. Date Last Written: 12/01/92 03:09:45
          Approx. Date Last Read: 01/31/94 00:40:16
          Number of Write Errors: 0
          Number of Read Errors: 0
          Last Update by (administrator):
          Last Update Date/Time: 01/02/93 13:20:14

```

Figure 48. Query Volume Command Detail Output

Use this report to ensure that the volume status is on-line and the access mode is read. Use the approximate date last written and approximate date last read fields to help you estimate the life of the volume. If more than six months have passed since the last time the volume has been read from, the volume is an excellent candidate to remove from the library.

Check storage volumes out of the library

To check storage volumes out of the library, start ADSM for AIX and the administrative client, if they are not already started. We recommend that you start an administrative client in console mode and another administrative client in interactive mode. You can review messages generated by the maintenance tasks in the console window. You can enter ADSM for AIX commands in the interactive window.

1. Login to AIX as the *root* user.
2. Start ADSM for AIX, if it is not already started. For example:

```
/usr/lpp/ars/bin/ars_adsm -sv
```

3. Start an ADSM for AIX administrative client in console mode. Enter the following command at the prompt of an aixterm window:

```
dsmadm -consolemode
```

4. Enter an ADSM for AIX administrator userid and password when prompted.
5. Start an ADSM for AIX administrative client in interactive mode. Enter the following command at the prompt of an aixterm window:

```
dsmdmc
```

6. Enter an ADSM for AIX administrator userid and password when prompted.
7. Make sure that there are no active ADSM for AIX sessions, other than the two administrator sessions that you started.
 - Issue the QUERY SESSION command to list the active ADSM for AIX sessions.
 - Issue the CANCEL SESSION command to stop an active ADSM for AIX session.
8. Issue the CHECKOUT LIBVOL command to remove a storage volume from a library. For example:

```
checkout libvol archlib0 avo11
```

Figure 49. Check Out Library Volume Command

Enter the command at the interactive client *adsm>* prompt. Replace the string *archlib0* with the name of the library. Replace the string *avo11* with the name of the storage volume that you want to check out.

9. After you reply to the confirmation message, ADSM for AIX changes the volume status to unavailable and ejects the optical platter from the library.
10. Check out as many storage volumes as required, to make room for new storage volumes. Repeat step 8.
11. When you have completed checking out storage volumes, close the ADSM for AIX administrative console and stop the ADSM for AIX client. Enter **QUIT** at the *adsm>* prompt to stop the client. Press the **Ctrl** and **C** keys to close the console.

Offline storage of optical platters

Refer to the documentation provided by the library manufacturer for instructions that describe how to handle optical platters and physically remove a platter from the library.

Refer to your organization's media storage guide for instructions about documentation you may need to complete when you remove a storage volume from a library and where to store optical storage volumes for safekeeping.

Chapter 12. Adding storage volumes for tape backup

During installation, you prepared tape storage volumes for backup of the ADSM database. After you archive report files into OnDemand, we recommend that you create a backup image of the ADSM database on one of the tape storage volumes that you prepared. Over time, you will use the initial set of tape storage volumes. You then need to initialize new tape storage volumes.

Stop OnDemand

Before you begin, we recommend that you stop the OnDemand system. Follow the instructions in “Stopping OnDemand” on page 36 to stop all of the processes related to OnDemand and remove mounted storage volumes from a library.

Label storage volumes

After stopping OnDemand and ADSM for AIX, complete the following steps to initialize tape storage volumes:

1. Login to AIX as the *root* user.
2. Change to the */usr/lpp/adsmserve/bin* directory.
3. Use the *dsmlabel* program to label storage volumes. For example:

```
dsmlabel -drive=/dev/mt0
```

Figure 50. Labeling Tape Storage Volumes

The example uses the default ADSM tape drive.

4. The *dsmlabel* program prompts you to place a blank tape in the drive.

Any tape with a pre-existing label is rejected and the next unlabeled tape is requested.

5. The *dsmlabel* program prompts you to type a label for the tape volume.

Labels can be from 1 to 6 alphanumeric characters in length. You can use the characters A through Z and 0 through 9. The label cannot include embedded blanks. ADSM for AIX converts lowercase letters in a volume label to uppercase.

We recommend that you maintain a central log with information about storage volumes. The log should contain information such as the date the volume was labeled, the type of media, the capacity, the volume label, and the type of data to be written on the storage volume.

6. After labeling the tape, ADSM for AIX ejects the tape from the drive.
7. Repeat step 4 to label additional tapes.
8. Press **Enter** to exit the *dsmlabel* program when you have completed labeling a set of tape storage volumes.

Restart OnDemand

When you have completed labeling storage volumes, you can restart OnDemand and ADSM for AIX. Follow the instructions in “Starting OnDemand” on page 33 to restart all of the processes related to OnDemand.

Chapter 13. Adding space to the ADSM database and recovery log

The ADSM database manages information about the location of OnDemand storage objects in storage pools and records information necessary for ongoing ADSM for AIX server operations. ADSM uses a recovery log to maintain a consistent database image by recording changes made to the database. Changes can include defining a policy domain or a storage pool and expiring storage objects.

During installation and configuration of the system, you defined and initialized logical volumes for the database and recovery log. You can increase the size of the database or recovery log by defining a new volume, determining how much the database or recovery log can be extended, and extending the capacity of the database or recovery log.

Define logical volumes

Before you can define a disk volume to ADSM for AIX for the database or recovery log, you must define a new logical volume for the database or recovery log. Complete the following steps to define a logical volume:

1. Login to AIX as the *root* user.
2. Define a logical volume using SMIT. Enter the following command at the AIX prompt:

```
smitty mklv
```

SMIT displays the Add a Logical Volume panel.

3. Press **F4** in the Volume Group Name field to list the volume groups defined to AIX. Select the name of the volume group where you want to add the new logical volume. For example, *arsvg* or *adsmvg*.

The *Introduction and Planning Guide* lists the recommended volume group name for the ADSM for AIX database and recovery log. Configuring ADSM for AIX in the *Installation and Configuration Guide* describes how to define the initial logical volumes for the database and recovery log.

4. Press **Enter**. SMIT displays the Add a Logical Volume panel.
5. Fill in the Logical Volume Name and Number of Logical Partitions fields for the database or recovery log. Table 2 lists examples of both. Press **Enter** after filling out each set of fields to define the logical volume.

Table 2. ADSM for AIX Logical Volumes

Purpose	Logical Volume Name	Number of Logical Partitions
Database	dsmdblv2	400
Recovery log	dsmloglv2	40

A formatted 2 GB fixed disk contains 479, 4 MB partitions. If you use the values in Table 2 on page 81, you will allocate almost the entire fixed disk to the ADSM for AIX logical volumes. The suggested values result in an allocation of approximately 1.6 GB for the database and 160 MB for the recovery log.

6. When complete, press **F10** to exit SMIT.

Define disk volumes

After you define new logical volumes for the database or recovery log, complete the following steps to define disk volumes for the database or recovery log. We recommend that you start an administrative client in console mode and another administrative client in interactive mode. You can review messages generated by the maintenance tasks in the console window. You can enter ADSM for AIX commands in the interactive window.

1. Start ADSM for AIX, if it is not already started. For example:

```
/usr/lpp/ars/bin/ars_adsm -sv
```

2. Start an ADSM for AIX administrative client in console mode. Enter the following command at the prompt of an aixterm window:

```
dsmdmc -consolemode
```

3. Enter an ADSM for AIX administrator userid and password when prompted.
4. Start an ADSM for AIX administrative client in interactive mode. Enter the following command at the prompt of an aixterm window:

```
dsmdmc
```

5. Enter an ADSM for AIX administrator userid and password when prompted.
6. Issue the DEFINE DBVOL command to define a disk volume for the database. For example:

```
define dbvolume dsbdb1v2
```

Enter the command at the *adsm>* prompt. Replace the string *dsbdb1v2* with the name of the logical volume that you defined for the database in "Define logical volumes" on page 81.

When the volume is defined, ADSM for AIX tracks the volume as part of the database logical volume. ADSM for AIX increases the space available for the database to include the size of the disk volume that you defined.

7. Issue the DEFINE LOGVOL command to define a disk volume for the recovery log. For example:

```
define logvolume dsmloglv2
```

Enter the command at the *adsm>* prompt. Replace the string *dsmloglv2* with the name of the logical volume that you defined for the recovery log (refer to “Define logical volumes” on page 81).

When the volume is defined, ADSM for AIX tracks the volume as part of the recovery log logical volume. ADSM for AIX increases the space available for the recovery log to include the size of the disk volume that you defined.

Determine the maximum extension

After you define volumes to the database or recovery log, complete the following steps to determine how much the database or recovery log can be extended. Query the database or recovery log to display the capacity and available space.

1. Start ADSM for AIX, if it is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start ADSM for AIX.
2. Start an ADSM for AIX administrative client in console mode, if one is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start an administrative client.
3. Start an ADSM for AIX administrative client in interactive mode, if one is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start an administrative client.
4. To determine the maximum extension of the database, enter the following command at the prompt of the interactive client:

```
query db
```

ADSM for AIX determines the maximum extension by subtracting the assigned capacity from the available space. The result is displayed in the *Maximum Extension* column in the report.

5. To determine the maximum extension of the recovery log, enter the following command at the prompt of the interactive client:

```
query log
```

ADSM for AIX determines the maximum extension by subtracting the assigned capacity from the available space. The result is displayed in the *Maximum Extension* column in the report.

Extend the capacity

Complete the following steps to extend the capacity of the database or recovery log. You must increase the capacity of the database or recovery log in 4 MB increments.

1. Start ADSM for AIX, if it is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start ADSM for AIX.
2. Start an ADSM for AIX administrative client in console mode, if one is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start an administrative client.
3. Start an ADSM for AIX administrative client in interactive mode, if one is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start an administrative client.
4. To increase the capacity of the database by 400 MB, enter the following command at the prompt of the interactive client:

```
extend db 400
```

5. To increase the capacity of the recovery log by 40 MB, enter the following command at the prompt of the interactive client:

```
extend log 40
```

6. ADSM for AIX displays status messages confirming successful completion of the command.

The space becomes available to ADSM for AIX without interrupting server operations.

Display the assigned capacity

After ADSM for AIX formats the new space for the database or recovery log, you can query the database or recovery log to display the capacity and available space.

1. Start ADSM for AIX, if it is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start ADSM for AIX.
2. Start an ADSM for AIX administrative client in console mode, if one is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start an administrative client.

3. Start an ADSM for AIX administrative client in interactive mode, if one is not already started. Refer to “Define disk volumes” on page 82 for instructions about how to start an administrative client.
4. To display the capacity of the database, enter the following command at the prompt of the interactive client:

```
query db
```

5. To display the capacity of the recovery log, enter the following command at the prompt of the interactive client:

```
query log
```

6. After reviewing the report, stop the administrative clients. Enter **QUIT** at the *adsm>* prompt to stop the interactive client. Press the **Ctrl** and **C** keys to close the administrative console.

Please refer to *Managing the Database and Recovery Log* in the *ADSM for AIX Administrator's Guide* for details about the ADSM database and recovery log.

Please refer to the *ADSM for AIX Administrator's Reference* for more information about using the ADSM for AIX administrative server and console and ADSM for AIX commands.

Chapter 14. Maintaining migrated index data

Migration is the process by which OnDemand moves index data from the database to archive media. This process optimizes database storage space while allowing you to maintain index data for a very long time. You typically migrate index data after users no longer need to access the reports, but, for legal or other business requirements, you still need to maintain the data for some number of years. If a user queries index data that has been migrated, an administrator must import a copy of the migrated table into the database. After maintaining the imported table in the database for a specified number of days, OnDemand expires it from the database.

This chapter provides information about importing index data into the database, including how to configure your system to migrate index data from the database to archive media, what happens when a user queries for migrated data, how to import the required index table, and what happens after you import a table into the database.

We assume that an experienced OnDemand administrator will use the information provided in this chapter. If you have questions about any of the following topics or if you would like help configuring your system to support migrating and importing of index data, please contact the IBM support center.

Configuring the system

Logging and routing messages

OnDemand provides the system logging facility so that administrators can monitor and manage the system. When you install and configure OnDemand, you initialize the system logging facility. The system logging facility is critical to the operation of the system.

When OnDemand processes a query for application group data requiring a table of index data that has been migrated to archive media, it writes an error message in the system log and sends the message to `/dev/console` and to the system log user exit.

You can use a facility that examines console messages and routes messages to an administrator, such as NetView. You can also configure the system log user exit to determine what action to take when a user queries migrated data.

Defining a system log user exit

Besides recording an “application group segment not available” error message in the system log, OnDemand sends the message to the system log user exit. If you have defined a system log user exit program, you can determine the action taken when OnDemand sends the message to the exit.

The program can take whatever action you deem necessary when the message occurs. For example, you may want to notify an administrator that a request for a migrated index table has occurred. Please refer to the *Installation and Configuration Guide* for

information about programming a response to messages sent to the system log user exit.

Configuring ADSM for AIX

Before OnDemand can migrate index data, you must configure ADSM for AIX to maintain the data on archive media. You must define the library where ADSM for AIX will maintain the index data, define a device class for the library, and configure a policy domain, including a storage pool. You must make sure that ADSM for AIX maintains data stored in the domain indefinitely. If you need to maintain a backup copy of the index data, we recommend that you implement a copy storage pool in ADSM for AIX. The *Installation and Configuration Guide* provides details about defining a library to ADSM for AIX and implementing a copy storage pool.

Defining storage sets

You must assign the System Migration application group to a storage set that writes data to archive media. The storage set must contain at least one primary storage node. If you need to maintain a backup copy of the index data on a different server than the primary storage node, you must define a secondary storage node. Use the OnDemand administrator interface to define storage sets and storage nodes and assign an application group to a storage set.

The storage nodes that you define must be registered as nodes in an ADSM for AIX domain. The domain must write the migration data to archive media and maintain the data indefinitely. If you define a secondary storage node, register it in a different domain than the primary storage node. Refer to the *Installation and Configuration Guide* for information about registering storage nodes in ADSM for AIX.

Configuring application groups

When you define an application group, you provide a set of storage management values that determine how long OnDemand maintains data stored in the application group and when OnDemand takes certain actions.

- Life of data and indexes

This value determines the length of time that OnDemand maintains index data and files stored in the application group.

- Length of time before migrating indexes to archive media

This value determines the number of days before OnDemand moves index data from the database to archive media.

Use caution when you configure application groups to migrate index data. The value that you specify should exceed the time frame that 99.9 percent of the accesses to the index data occurs. In other words, only in exceptional situations should users need to access data in migrated tables. If a user needs index data that has been migrated to archive media, the process of importing the table into the database requires administrator intervention, and usually results in a significant delay in completing the query. Also, additional space in the database must be available for imported index tables.

- Length of time to keep imported migrated indexes

This value determines how long OnDemand maintains imported index data in the database before it is scheduled for deletion.

If you need to maintain index data on archive media, you must provide a value for the length of time before migrating indexes to archive media. If you want OnDemand to maintain imported index data in the database for longer than 30 days, you must provide a value for the length of time to keep imported migrated indexes. OnDemand deletes imported index data after it resides in the database for the number of days specified in the length of time to keep imported migrated indexes or the life of data and indexes, whichever occurs first.

Use the OnDemand administrator interface to define application groups, including configuring storage management options to maintain index data.

What happens when a user queries migrated data

Message to the user

When OnDemand needs to access migrated index data to complete a query, it sends a message to the client program. The message states that the data required to complete the query is not available and the user should contact an administrator.

Message to the system log

When OnDemand needs to access migrated index data to complete a query, it writes a message to the system logging facility. You can open the System Log folder to display messages written to the system logging facility.

Message to */dev/console*

When OnDemand needs to access migrated index data to complete a query, it sends a message to */dev/console*. You can use a facility that examines console messages and routes messages to an administrator, such as NetView.

Message to the system log user exit

When OnDemand needs to access migrated index data to complete a query, it sends a message to the system log user exit. If you do not use some other facility that routes messages to an administrator (such as NetView), we strongly encourage you to implement a user-defined program to process the message. The function of the program can vary, however, at a minimum, we recommend that the program notifies a system administrator when a query for migrated index data occurs.

Importing the index data

Verify temporary work space

Importing migrated index data into the database requires temporary work space. Before you import the index data, you should verify that sufficient free space is available in the directory or file system that OnDemand uses for temporary work file storage. The ARS_TMP environment variable determines where OnDemand stores temporary work files. The ARS_TMP environment variable is set in the ars.cfg file.

Verify database storage space

Importing migrated index data into the database requires additional database storage. Before you import the index data, you should verify that sufficient free space is available in the database directories.

Verify database log file space

Importing migrated index data into the database requires DB2 for AIX log file storage. Before you import the index data, you should verify that sufficient free space is available in the archive log file directory. The ARS_ARCHIVE_LOGPATH environment variable determines the location of the archive log files. The ARS_ARCHIVE_LOGPATH environment variable is set in the ars.cfg file.

Run the arsadmin command

OnDemand provides the arsadmin command to import a table of index data into the database. The following example shows how to use the import command function to copy a migrated index table from archive media into the database. The name of the application group and the index table to import can be obtained from the message written to the system log.

```
/usr/lpp/ars/bin/arsadmin import -g 'Credit'  
-u admin -p "" SL27
```

Refer to Chapter 19, “arsadmin command reference” on page 117 for a description of the arsadmin command and details about parameters and options.

After you import index data

Querying data

After you import the required index data into the database, you should notify the user to re-enter the query.

Expiring imported migrated indexes

OnDemand flags an imported index table for deletion after it resides in the database for the number of days specified in the length of time to keep imported indexes field. After reaching this value, the next time that the arsmaint command runs, the data is deleted.

You typically configure the `arsmaint` command to run automatically on a regular schedule, using the cron facility. The *Installation and Configuration Guide* provides details.

If you need to run the `arsmaint` command on an exception basis, you can start it from the AIX prompt. Chapter 23, “`arsmaint` command reference” on page 149 provides details.

Verify the length of time before migrating indexes to archive media

If you find that queries for migrated index data occur often, we recommend that you update the application group and increase the length of time that OnDemand maintains index data in the database. This should reduce the number of queries that need migrated index data and later importing of data.

Verify the length of time to keep imported migrated indexes

OnDemand deletes imported index data after it resides in the database for the number of days specified in the length of time to keep imported migrated indexes or the life of data and indexes, whichever occurs first.

The default value for the length of time to keep imported migrated indexes is 30 days. If you want OnDemand to maintain imported index data in the database for longer than 30 days, you must update the application group and change the length of time to keep imported migrated indexes.

If you need to maintain application group data longer than the current life of data and indexes, you must update the application group and change the life of data and indexes.

Part 3. Command Reference

This section of the book contains reference information about the OnDemand administrative commands. Please read the "Overview," beginning on page 95 for important information about interpreting command syntax.

Topic	Page
ars_adsm command reference	99
ars_db command reference	103
ars_load command reference	111
arsadmin command reference	117
arsdate command reference	129
arsdocget command reference	133
arsjesd command reference	145
arsmaint command reference	149
arsquery command reference	155
arstblsp command reference	161
arsuser command reference	165
arsview command reference	175

Chapter 15. Overview

The commands contained in this reference include general OnDemand system commands and commands that control ADSM for AIX and DB2 for AIX. The commands are presented in alphabetical order. Each command contains a description of its purpose and syntax (including descriptions of the parameters that can be used). Examples and general information about using the commands have also been included.

Examples of commands may appear in one of the following two formats:

```
/usr/lpp/ars/bin/arsmaint -cdmsv >> /tmp/arsmaint.log 2>&1  
  
/usr/lpp/ars/bin/arsmaint -cdmsv  
  >> /tmp/arsmaint.log 2>&1
```

However, when you type the command, you must enter the entire command syntax on a single line.

How to read a syntax diagram

A syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type.

Read a syntax diagram from left to right and from top to bottom, following the horizontal line (the main path). If the line ends with an arrowhead, the command syntax is continued and the next line starts with an arrowhead. Facing arrowheads mark the end of the command syntax.

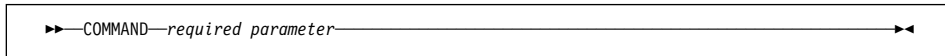
When you type a command from the syntax, be sure to include punctuation, such as commas and equal signs.

Parameters are classified as keywords or variables:

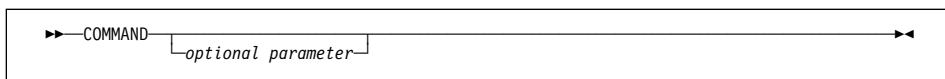
- Keywords represent constants and are shown (in syntax) in uppercase letters; however, at the command prompt, you can enter keywords in either uppercase or lowercase. An example of a keyword is a command name. AIX and OnDemand command names are case sensitive.
- Variables represent names or values you supply and are shown (in syntax) in lowercase letters; however, at the command prompt, you can enter variables in either uppercase or lowercase unless the syntax directions explicitly state the case restrictions. An example of a variable is a file name. AIX and OnDemand file names are case sensitive.

A parameter can be a combination of a keyword and a variable.

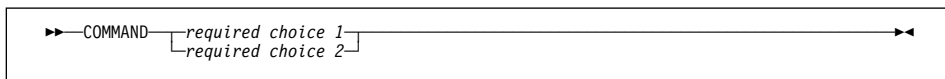
Required parameters are displayed on the main path.



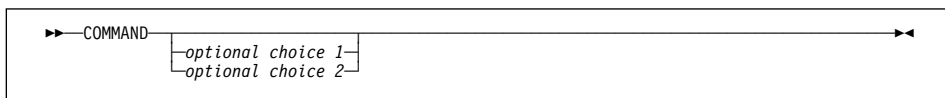
Optional parameters are displayed below the main path.



A stack of parameters, with the first parameter displayed on the main path, shows that you must choose one of the parameters.

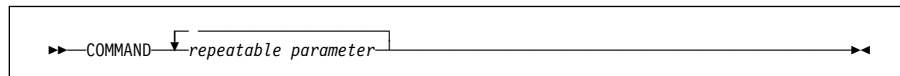


A stack of parameters, with the first parameter displayed below the main path, shows that you can choose one of the parameters.

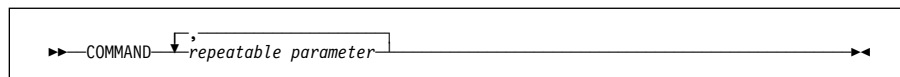


An arrow returning to the left, above the path, shows that items can be repeated following these conventions:

- If the repeat arrow contains a break, the item can be repeated in a list with the items separated by blank spaces.



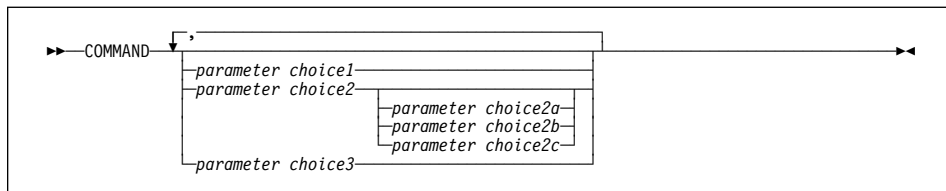
- If the repeat arrow contains a comma, the item can be repeated in a list with the items separated by commas.



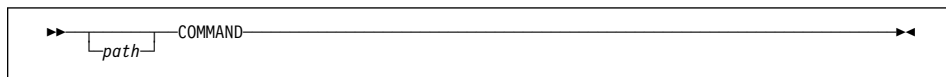
You can repeat items from parameter stacks following the stack conventions for required and optional parameters described previously.

Some syntax diagrams contain parameter stacks within other parameter stacks. You can only repeat items from stacks according to the conventions described previously. That is, if an inner stack does not have a repeat arrow above it but an outer stack does, you can choose only one parameter from the inner stack and combine it with any parameter from the outer stack, and that combination can be repeated. For example,

the following diagram shows that you could combine parameter choice2a with parameter choice2 and then you can repeat that combination again (choice2 plus choice2a).

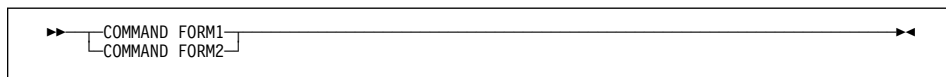


Some commands are preceded by an optional path parameter.



If you do not supply the path parameter, the system searches the current directory for the command. If the command is not in the current directory, the system continues to search for the command using the directories defined in the PATH environment variable.

Some commands in this section have several formats that accomplish the same task. These commands appear (in syntax) similar to the following:



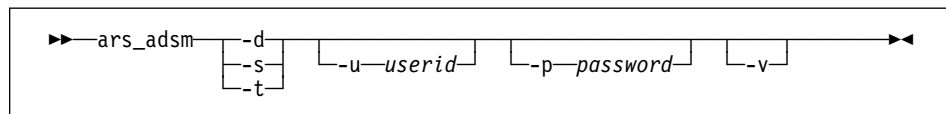
The description of the command directs you to the correct format to use.

Chapter 16. ars_adsm command reference

Purpose

Provides a command line interface for specific ADSM for AIX functions, such as starting and stopping the ADSM for AIX server and creating a backup image of the ADSM database.

Syntax



Description

The `ars_adsm` command provides a command line interface to ADSM for AIX for the following functions:

- Start the ADSM for AIX server as a background process.
- Stop the ADSM for AIX server.
- Create a backup image of the ADSM database.

When you perform ADSM for AIX administrative tasks, you must supply the ID and password of an ADSM administrative user. A set of default values for the `userid` and `password` are coded in the `ars_adsm` shell script. You can use the values in the script, change the values in the script, or specify the values using the `-u` and `-p` command line parameters. The *Installation and Configuration Guide* describes how to change the ADSM administrative `userid` and `password` values in the script.

Parameters

- d** Create a backup image of the ADSM database. Writes the backup image to the device that you configured when you installed and configured ADSM for AIX. The *Installation and Configuration Guide* provides details about defining the database backup device.

If you plan to create a backup image of the database on tape, you must mount a labeled, ADSM for AIX tape storage volume in the tape drive before the `ars_adsm` command starts. Chapter 12, “Adding storage volumes for tape backup” on page 79 provides details about labeling tape volumes that can be used when you backup the ADSM database.
- s** Start the ADSM for AIX server as a background process.

- t** Stop the ADSM for AIX server.
- u userid**
An ADSM for AIX administrative userid.
- p password**
The password for the ADSM administrative user.
- v** Verbose mode. Print messages to standard output.

Examples

1. To start the ADSM for AIX server, type the following command at the prompt:

```
/usr/lpp/ars/bin/ars_adsm -sv
```

The example uses the administrative userid and password set in the shell script.

2. To stop the ADSM for AIX server, type the following command at the prompt:

```
/usr/lpp/ars/bin/ars_adsm -tv
```

The example uses the administrative userid and password set in the shell script.

3. To create a backup image of the ADSM database, type the following command at the prompt:

```
/usr/lpp/ars/bin/ars_adsm -dv
```

The example uses the administrative user ID and password set in the script.

4. The following example shows a crontab entry that causes the cron program to automatically start the `ars_adsm` command and create a backup copy of the ADSM database every night at 11 p.m.

```
00 23 * * * /usr/lpp/ars/bin/ars_adsm -dv  
>> /tmp/ars_adsm.log 2>&1;
```

The example uses the administrative user ID and password set in the script.

Notes

OnDemand server programs can remain active while you create a backup copy of the ADSM database.

If you stop ADSM for AIX while the OnDemand server daemon is active, OnDemand issues an error message, if an end-user attempts to retrieve an item that resides on a storage volume managed by ADSM for AIX.

Files

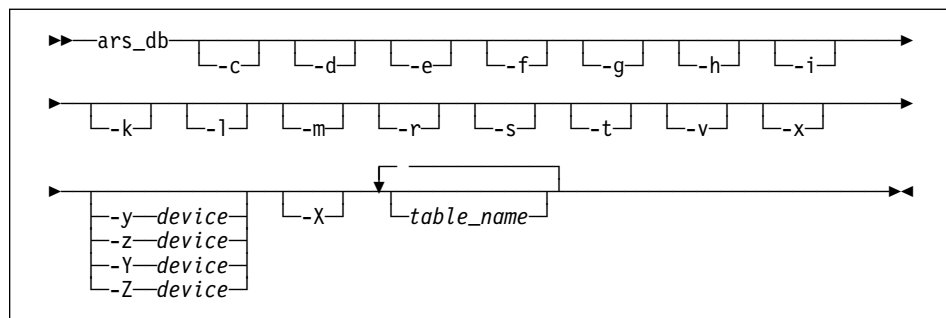
<code>/usr/lpp/ars/bin/ars_adsm</code>	The executable program.
--	-------------------------

Chapter 17. ars_db command reference

Purpose

Provides a command line interface for specific OnDemand database functions, such as creating the database, starting and stopping DB2 for AIX, creating a backup image of the database, and reorganizing OnDemand system tables. The OnDemand database resides on the library server.

Syntax



Description

The `ars_db` command is a Korn shell script that provides an interface to DB2 for AIX for the following database functions:

- Create and initialize the OnDemand database.
- Start DB2 for AIX.
- Stop DB2 for AIX.
- Create backup images of the OnDemand database. You can use the `ars_db` command to create backup images of table spaces and the full database. In addition, the `ars_db` command can use ADSM to manage the backup images. However, before you use the `ars_db` command to backup images to ADSM-managed storage, you must properly configure ADSM. Refer to the *Installation and Configuration Guide* for details.
- Reorganize and optimize OnDemand system tables.
- Import and export tables and databases.

The `ars_db` script contains DB2 for AIX variables that support the OnDemand database environment. Before you run the script, you must verify, and if necessary, change the settings in the script. The *Installation and Configuration Guide* provides details about script variables you may need to change to support your OnDemand environment.

Parameters

- c** Create the (initial) OnDemand database structure.

The **-c** parameter implies the **-r**, **-s**, and **-t** parameters.

The `ars_db` command reads the `ars.cfg` file to obtain the database configuration information. Before creating the database, verify and if necessary, make changes to the settings in the file. The *Installation and Configuration Guide* provides information about the database configuration variables you may need to change to support your OnDemand system.
- d** Drop tables from the database. The default is all tables. To drop a specific table, use the **-d tablename** form of the command, where *tablename* is the name of the OnDemand system table.
- e** Drop indexes from the database. The default is all indexes. To drop a specific index, use the **-e tablename** form of the command, where *tablename* is the name of the OnDemand system table.
- f** Ignore DB2 for AIX failures.
- g** Start DB2 for AIX.
- h** Stop DB2 for AIX.
- i** Import tables. The default is all OnDemand system tables. To import a specific table, use the **-i tablename** form of the command, where *tablename* is the name of the OnDemand system table and the name of the file that contains table to be imported. OnDemand reads the file from the directory where the `ars_db` command was invoked.
- k** Activate the connection to the OnDemand database.
- l** Import or export tables in the IXF file format instead of the DEL file format (the default).

The IXF file format is the PC version of the Integrated Exchange Format. A DEL file is a sequential ASCII file with row and column delimiters.

This flag is valid only with the **-i** and **-x** flags.
- m** Perform maintenance on the OnDemand database, reorganizing the OnDemand system tables. This option refreshes the tables and optimizes access to information in the database. The system must be idle when you run the `ars_db` command with this option.
- r** Create configuration indexes. The default is all indexes. To configure a specific index, use the **-r tablename** form of the command, where *tablename* is the name of the OnDemand system table.
- s** Run database statistics. This parameter is used to optimize indexes and tables and make access to information as efficient as possible. The default is all OnDemand system tables. To run statistics for a specific table, use the **-s tablename** form of the command, where *tablename* is the name of the OnDemand system table.

- t** Create configuration tables. The default is all OnDemand system tables. To configure a specific table, use the *-t tablename* form of the command, where *tablename* is the name of the OnDemand system table.
- v** Verbose mode. Write processing and status messages to *stdout*.
- x** Export tables. The default is all OnDemand system tables. To export a specific table, use the *-x tablename* form of the command, where *tablename* is the name of the OnDemand system table and the name of the file OnDemand creates. OnDemand writes the file in the directory where the *ars_db* command was invoked.

-y device

Create an offline backup image of the OnDemand database on the specified device.

Before you start an offline backup of the database, you must make sure that no other users or processes are connected to the OnDemand database. We recommend that you run the offline backup at a time when you are certain no users will access OnDemand or shutdown the OnDemand system before you start the offline backup.

If you specify a tape device, you must place a blank tape volume in the device before starting the *ars_db* command.

If the specified device is ADSM, it means that ADSM for AIX manages the database backup image. However, before you use the *ars_db* command to backup images to ADSM-managed storage, you must properly configure ADSM. Refer to the *Installation and Configuration Guide* for details. The *DB2 for AIX Administration Guide* also contains important information about ADSM for AIX support and applicability for your system.

-z device

Create an online backup image of the OnDemand database on the specified device.

An online backup of the database can run while other users and processes are connected to the database. That is, other users and applications can read and modify the database while the backup process is running.

If you specify a tape device, you must place a blank tape volume in the device before starting the *ars_db* command.

If the specified device is ADSM, it means that ADSM for AIX manages the database backup image. However, before you use the *ars_db* command to backup images to ADSM-managed storage, you must properly configure ADSM. Refer to the *Installation and Configuration Guide* for details. The *DB2 for AIX Administration Guide* also contains important information about ADSM for AIX support and applicability for your system.

- X** Do not backup the USERSPACE1 table space. The USERSPACE1 table space is created by DB2 when you initialize the OnDemand database. It contains system tables and application group tables that have not been migrated to their own table spaces. This parameter is only valid with the *-Y* and *-Z* flags.

A typical use of this parameter is to test the table space backup process. For example, assume that you migrated a table of application group data to its own table space. All other tables of application group data remain in the USERSPACE1 table space. To create an online backup image of the migrated application group table, run the `ars_db` command with the `—XZ` parameters.

-Y device

Create an incremental, offline backup image of table spaces on the specified device. An incremental backup means that OnDemand backs up only those tables that have changed since the last time they were backed up. OnDemand keeps track of table space backups and can determine which table spaces have changed and need to be backed up.

Before you start an offline backup of the database, you must make sure that no other users or processes are connected to the OnDemand database. We recommend that you run the offline backup at a time when you are certain no users will access OnDemand or shutdown the OnDemand system before you start the offline backup.

If you specify a tape device, you must place a blank tape volume in the device before starting the `ars_db` command.

If the specified device is ADSM, it means that ADSM for AIX manages the backup image. However, before you use the `ars_db` command to backup images to ADSM-managed storage, you must properly configure ADSM. Refer to the *Installation and Configuration Guide* for details. The *DB2 for AIX Administration Guide* also contains important information about ADSM for AIX support and applicability for your system.

-Z device

Create an incremental, online backup image of table spaces on the specified device. An incremental backup means that OnDemand backs up only those tables that have changed since the last time they were backed up. OnDemand keeps track of table space backups and can determine which table spaces have changed and need to be backed up.

An online backup of the database can run while other users and processes are connected to the database. That is, other users and applications can read and modify the database while the backup process is running.

If you specify a tape device, you must place a blank tape volume in the device before starting the `ars_db` command.

If the specified device is ADSM, it means that ADSM for AIX manages the backup image. However, before you use the `ars_db` command to backup images to ADSM-managed storage, you must properly configure ADSM. Refer to the *Installation and Configuration Guide* for details. The *DB2 for AIX Administration Guide* also contains important information about ADSM for AIX support and applicability for your system.

table_name

Optionally specify the name of one or more OnDemand system tables for the drop (-d and -e), import (-i), create configuration (-r and -t), run statistics (-s), and

export (-x) parameters. If you do not specify the **tablename** parameter, OnDemand runs the command for all OnDemand system tables.

For the import (-i) parameter, OnDemand reads the imported tables from a file in the directory where *ars_db* was invoked. For the export (-x) parameter, OnDemand writes the exported tables to a file in the directory where *ars_db* was invoked.

Examples

1. The following example shows how to start DB2 for AIX from the command line.

```
/usr/lpp/ars/bin/ars_db -gkv
```

2. The following example shows how to stop DB2 for AIX from the command line.

```
/usr/lpp/ars/bin/ars_db -hv
```

3. The following example shows how to reorganize all the OnDemand system tables in the database and run the database statistics command, which optimizes access to the tables.

```
/usr/lpp/ars/bin/ars_db -msv
```

4. The following example shows how to reorganize a specific OnDemand system table. In the example, the *ars_db* command reorganizes the indexes for the Application Group table.

```
/usr/lpp/ars/bin/ars_db -mv arsag
```

5. The following example shows how to export a specific OnDemand system table to an IXF format file.

```
/usr/lpp/ars/bin/ars_db -xl arsag
```

6. The following example shows how to create an offline backup image of the OnDemand database on a tape volume. You must place a tape in the device before you start the *ars_db* command.

```
/usr/lpp/ars/bin/ars_db -v -y /dev/rmt0
```

7. The following example shows a crontab entry that causes the cron program to run the `ars_db` command and create an online backup copy of the OnDemand database every night at 9:30 p.m. You must mount a tape in the device before cron starts the `ars_db` command.

```
30 21 * * * /usr/lpp/ars/bin/ars_db -v -z /dev/rmt0  
>> /tmp/ars_db.log 2>&1;
```

8. The following example shows how to create an online, incremental backup of table spaces in the OnDemand database. The backup image is written to storage managed by ADSM.

```
/usr/lpp/ars/bin/ars_db -v -Z ADSM
```

9. The following example shows how to create an offline, incremental backup of table spaces in the OnDemand database. The backup image is written to storage managed by ADSM.

```
/usr/lpp/ars/bin/ars_db -v -Y ADSM
```

10. The following example shows how to create an online, full backup of the OnDemand database. The backup image is written to storage managed by ADSM.

```
/usr/lpp/ars/bin/ars_db -v -z ADSM
```

11. The following example shows how to create an offline, full backup of the OnDemand database. The backup image is written to storage managed by ADSM.

```
/usr/lpp/ars/bin/ars_db -v -y ADSM
```

Notes

- The `ars_db` command should be run only on the library server.
- Stop OnDemand or the server daemon before you run the `ars_db` command with the create (-c), import (-i), reorganize (-m) or offline database backup (-y, -Y)

parameters. “Stopping OnDemand” on page 36 provides details about stopping OnDemand and the server daemon.

- The OnDemand server daemon can remain active if you need to stop DB2 for AIX. However, OnDemand records an error message in the system log and the OnDemand client program issues an error message whenever DB2 for AIX is stopped and an end-user attempts to query the database.
- The online database backup command (-z, -Z parameters) can be run while other applications are connected to the database.
- We recommend that you do not run the ars_load command at the same time that you run the ars_db command with the -m or -s options.

Files

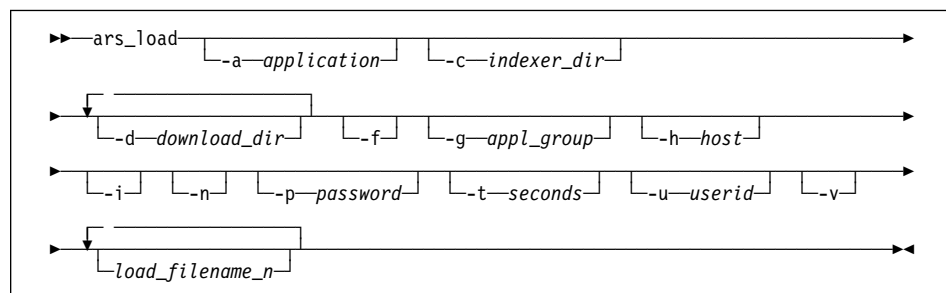
<code>/usr/lpp/ars/bin/ars_db</code>	The executable program.
--------------------------------------	-------------------------

Chapter 18. ars_load command reference

Purpose

The `ars_load` command can be used to index report files and load data into an application group. The `ars_load` command determines if the input file needs to be converted or indexed and if so, calls the conversion and indexing program. The `ars_load` command then calls the programs that load report files on storage volumes and update the OnDemand database with the index information extracted from the report file.

Syntax



Description

The `ars_load` command is the main OnDemand data indexing and loading program. The `ars_load` command calls the data conversion and indexing program if the data needs to be indexed, creates input files for OnDemand, and calls programs that load data on storage volumes and update the OnDemand database with index data. The `ars_load` command writes processing messages to the system logging facility. You can open the System Log folder and review the messages generated during a report file load.

You typically set up the `ars_load` command to run as an AIX daemon, periodically checking specific file systems for input data downloaded from other systems and automatically processing the downloaded files.

When you run the `ars_load` command, you must supply the `userid` and `password` of an OnDemand user with administrator authority for the application group. A set of default values for the `userid` and `password` are coded in the `ars_load` shell script. You can use the default values, change the values in the script, or specify the values at run time using the `-u` and `-p` parameters. The *Installation and Configuration Guide* provides details about changing the values in the script.

Parameters

-a application

The name of the application within the application group. If the application group to load contains more than one application, you must specify this flag and the application name.

-c indexer_dir

The file system where OnDemand temporarily stores files created by the indexing program. The default is the directory where the `ars_load` command was invoked.

For better performance, we recommend the **-c indexer_dir** point to a different file system than the **-d download_dir** parameter.

-d download_dir

The file system that contains report data transferred from another system. For example, you can specify the name of the file system where the `arsjesd` command stores data downloaded from an MVS system.

You can specify this parameter and value more than once, which causes the `ars_load` command to check multiple download directories.

For better performance, we recommend the **-d download_dir** parameter point to a different file system than the **-c indexer_dir** parameter.

MVS Download uses the following file naming convention:

```
HLQ.JOBNAME.DATASET.FORMS.YYDDD.HHMMSS.ARD
```

- The `FORMS` part of the filename can be used to identify the application group to load. Optionally, you can use the **-g** flag to identify the application.
- If the application group to load contains more than one application (source of data), the `DATASET` part of the filename can be used to identify the application within the application group. Optionally, you can use the **-a** flag to identify the application.

-f Unload the data, if the report file load fails. If either the database manager or the storage manager step fails, OnDemand deletes any index data added to the database and storage objects placed on storage volumes during the report file load.

If the report file load fails, check the message log in the system logging facility. If the message log contains a Load ID, it means OnDemand stored some of the report file data in the application group. Before you restart the report file load, you must unload the data with the `arsadmin` command. Chapter 19, “arsadmin command reference” on page 117 provides details about using the `arsadmin` command.

-g appl_group

The name of the application group. This parameter is required, if you specify the `load_filename_n` parameter. This parameter is optional, if you specify the **-d**

download_dir parameter. If you specify the **-d download** parameter, by default, the `ars_load` command uses the `FORMS` part of the downloaded filename to determine the name of the application group to load.

-h host

The fully qualified host name or IP address of the OnDemand library server.

You must specify this flag and the library server name, when you run the `ars_load` command on an OnDemand object server that resides on a different RS/6000 than the library server.

-i Run the data indexing program only, do not load report file data on storage volumes or indexing information in the database.

-n Determines whether OnDemand deletes the input files when the `ars_load` command ends.

- If you specify **-n**, OnDemand does not delete the input files when the `ars_load` command ends.
- If you do not specify **-n**, OnDemand deletes the input files when the `ars_load` command ends.

In either case, if the `ars_load` command fails in the load step because of a device or system problem, you can restart the load step after correcting the problem, using the intermediate files created by the `ars_load` command. The `ars_load` command stores the intermediate files in the work directory named with the **-c** flag or the directory where you started the `ars_load` command, if you did not specify the **-c** flag. The intermediate files have the same filename as the original input file.

-p password

The password for the user specified with the **-u** parameter. If you specify the **-p** parameter, you must specify the **-u** parameter.

-t seconds

Determines the polling time in seconds. This is the interval that the `ars_load` command checks the data download directories. The default is 600 seconds (ten minutes).

-u userid

The userid of an OnDemand user with administrator authority for the application group. The user must have permission to add documents to the application group. If you specify the **-u** parameter, you must specify the **-p** parameter.

-v Write processing messages to `stdout`.

load_filename_n

Determines the filename of one or more input files to process. The `ars_load` command concatenates the following filetype extensions with the filename that you specify: **.out**, **.ind**, and **.res**.

- An **.out** file contains the report data.
- An **.ind** file contains the index data.
- An **.res** file contains the resource data.

If the `ars_load` command does not find a file with the `.ind` filetype extension, it automatically calls the `arsacif` command to index the input file.

If you do not specify a filename, the `ars_load` command checks the directory specified with the `-d` flag for input files with the `.ARD` filetype extension.

Examples

1. In the following example, the `ars_load` command is invoked from the AIX prompt to check for data transmitted by MVS Download in the specified download directory. If the data needs to be indexed, the `ars_load` command calls the `arsacif` command. The `ars_load` command stores work files in the specified directory. The `ars_load` command uses the `userid` and `password` found in the shell script to login to OnDemand and access the application group. OnDemand expects to find files with the type of `ARD` in the download directory. The `ars_load` command extracts the name of the application group to load from the input filename.

```
/usr/lpp/ars/ars_load -d /arsacif/acif1 -c /arsacif/acif2
```

2. In the following example, the `ars_load` command is invoked from the AIX prompt to store data in the specified application group. If the data needs to be indexed, the `ars_load` command calls the `arsacif` command. The `userid` and `password` of a user with administrator authority for the application group is specified. An input filename is specified.

```
/usr/lpp/ars/ars_load -g BILLS -p"" -u admin P03510
```

3. In the following example, the `ars_load` command is invoked from the AIX prompt to store data in the specified application group. The application group name contains an embedded blank character, and must be quoted. If the data needs to be indexed, the `ars_load` command calls the `arsacif` command. The `userid` and `password` of a user with administrator authority for the application group is specified. Two input filenames are specified.

```
/usr/lpp/ars/ars_load -g "ABC Credit" -p"" -u admin RW7505 RW8505
```

4. In the following example, the `ars_load` command is invoked from the AIX prompt to store data in the specified application group. The IP address on the OnDemand library server is specified. This format of the command can be used to load data on an object server and store index data on the library server. If the data needs to be indexed, the `ars_load` command calls the `arsacif` command. The `userid` and `password` of a user with administrator authority for the application group is specified. An input filename is specified.

```
/usr/lpp/ars/ars_load -g "BILLS" -h "9.99.111.222"  
-p"" -u admin ARSP0MST
```

Notes

We recommend that you do not run the `ars_load` command at the same time that you run database statistics (for example, running the `arsmaint` command with the `-r` option).

We recommend that you do not run the `ars_load` command at the same time that you run the `ars_db` command with the `-m` or `-s` options.

Files

<code>/usr/lpp/ars/bin/ars_load</code>	The executable program.
--	-------------------------

Chapter 19. arsadmin command reference

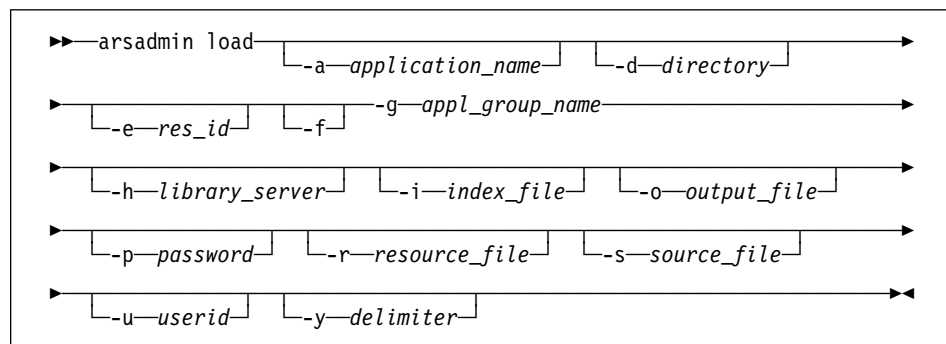
Purpose

The arsadmin command is the primary OnDemand data processing program. The ars_load command calls the arsadmin command to process report, index, and resource files and create index data and storage objects. The import command function can be used to copy migrated index data from archive media to the database. The arsadmin command can also be invoked from the AIX prompt by the experienced OnDemand administrator to help with problem determination and recovery.

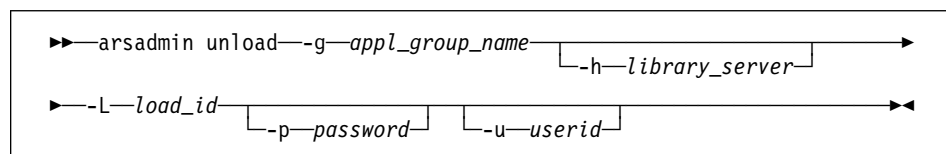
Syntax

The arsadmin command provides the following functions:

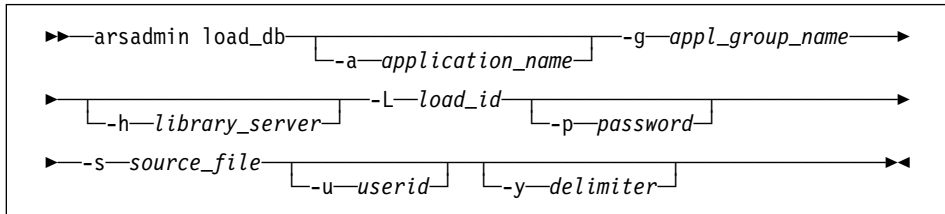
Load command function



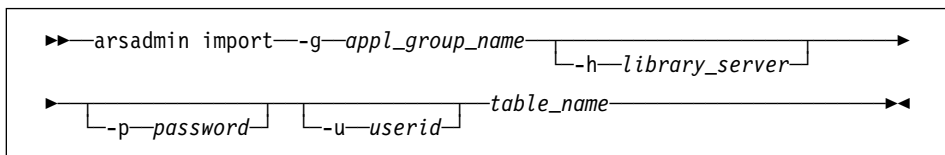
Unload command function



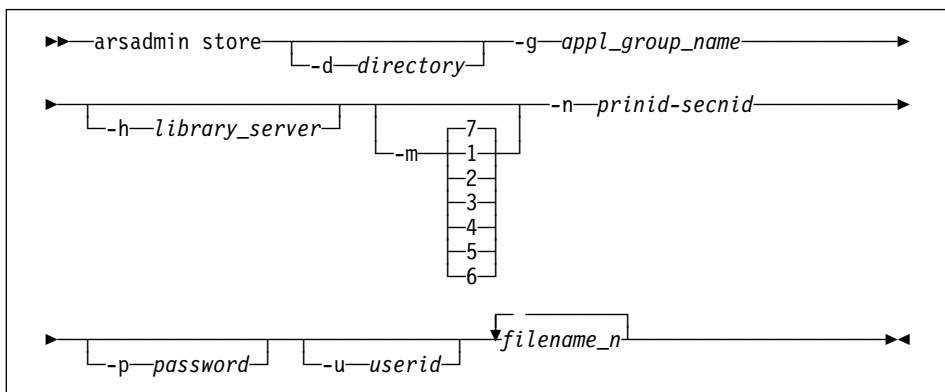
Load_db command function



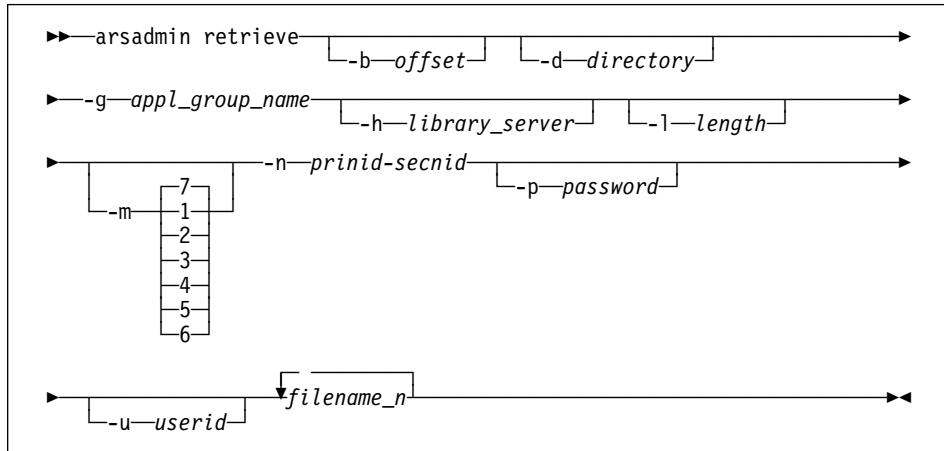
Import command function



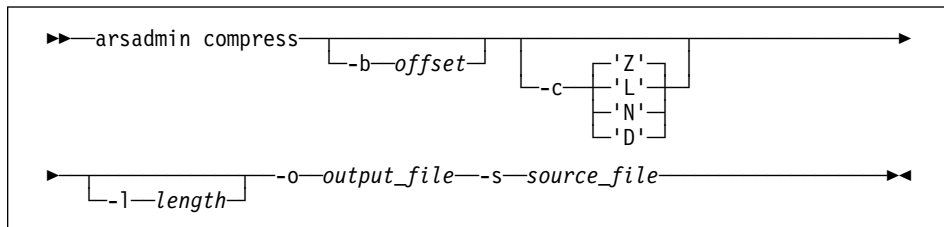
Store command function



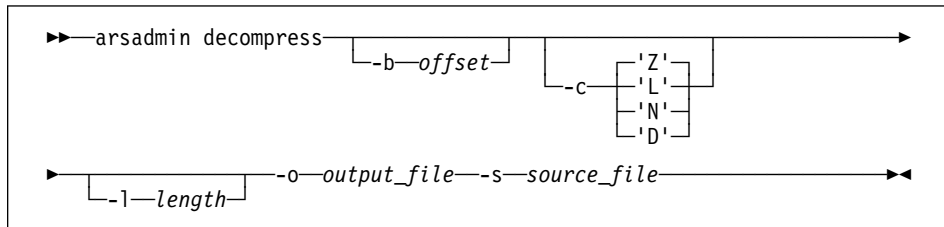
Retrieve command function



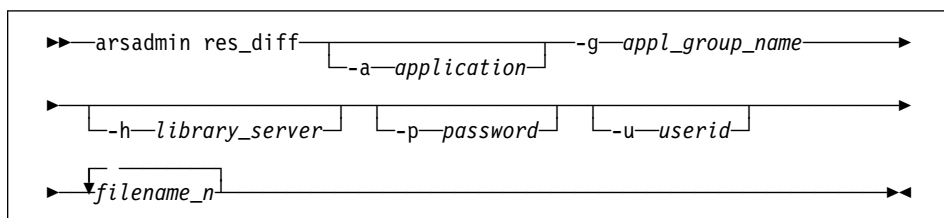
Compress command function



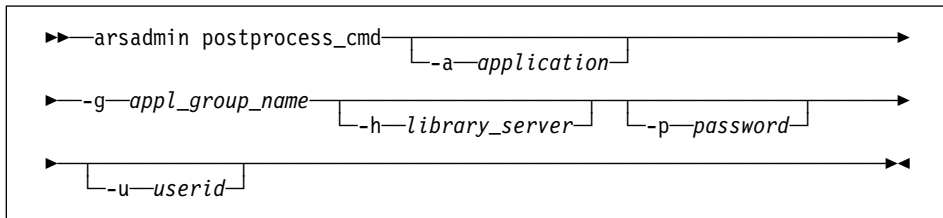
Decompress command function



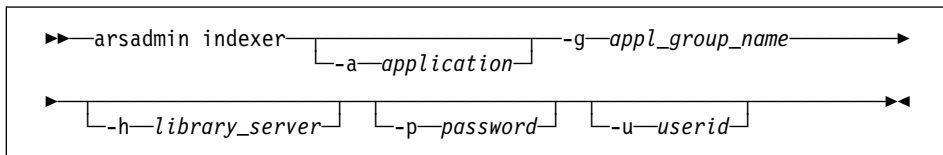
Res_diff command function



Postprocess_cmd command function



Indexer command function



Description

The `arsadmin` command processes the output files produced by the data indexing program (report files, index files, and optionally, resource group files), to load index data into the OnDemand database and store copies of the report files and resources in cache storage and on archive media.

The import command function can be used to import migrated index data from archive media to the database.

The `arsadmin` command can also be used by the OnDemand administrator for testing, problem determination, and data recovery. The experienced OnDemand administrator can create, delete, and restore storage objects, retrieve individual report files and resources from storage volumes, and calculate data compression ratios.

The `arsadmin` command provides the following functions:

load The load command function creates the storage objects used by the OnDemand storage manager to load data in cache storage and on archive media and passes index data and parameters to the database manager to update the OnDemand database. Input data can be one or more of the following: a report file (**-s source_file**), an index file (**-i index_file**), or a resource group file (**-r resource_file**).

If you use ACIF to index a report file, you can concatenate the three types of input data into a single file. If you do so, the **-s**, **-i**, and **-r** parameters must specify the same file name.

unload The unload command function deletes the storage objects and database rows associated with a Load ID from OnDemand. A Load ID can identify one or more storage objects. You would typically use this command function to

delete the Load ID associated with a report file that you want to back out of OnDemand. You can also use this command function to delete a Load ID created during testing. The userid that you specify when you run the unload command function must have permission to delete documents from the application group.

- load_db** The `load_db` command function stores index data in the OnDemand database. If you specify postprocessing instructions for an application, the `ars_load` command calls the `load_db` command function to store the index data in the database, after postprocessing the index file created by the `load` command function. Index data created with the `load` command function can be saved to a file.
- import** The `import` command function can be used to import migrated index data from archive media to the database. When OnDemand determines that a query requires index data that has been migrated to archive media, a message is written to the system log. An administrator must use the `arsadmin` command to import the required table into the database. After the index data has been imported, the user can re-enter the query. Importing migrated index data requires additional space in the database and DB2 for AIX log file storage.
- store** The `store` command function causes the OnDemand storage manager to store objects contained in a file into an application group. You can specify a mask that causes the storage manager to store the objects on one or more storage locations, that is, the primary storage node, secondary storage node, and cache storage. You would typically use this command function to copy storage objects from one location to another, for example, copy a storage object previously loaded in cache storage to archive media.
- retrieve** The `retrieve` command function causes the OnDemand storage manager to retrieve a storage object from the specified location. You can specify a mask that tells the storage manager where to retrieve the storage object from, that is, the primary storage node, secondary storage node, or cache storage. After retrieving the storage object, you can use the `decompress` command function to extract a specific report file or resource from the storage object file.
- compress**
- The `compress` command function can be used to compress report files and resources in preparation for creating storage objects. The `compress` command function can also be used to estimate how much storage space will be required to store a report file in OnDemand
- To estimate the amount of storage required to store a report file that contains logical items, such as bills, invoices, or statements, compress a portion of the report file (with the `-l` parameter). Divide the size of the output (compressed) file by the number of bytes of report file data that you compressed. The result is the compression ratio achieved.
- To estimate the amount of storage required to store a report file that contains line data, such as a transaction log, extract a portion of the report file to a separate file and compress the segment of the report file. Divide the size of

the output (compressed) file by the size of the segment of the report file. The result is the compression ratio achieved.

decompress

The decompress command function can be used to extract a specific report file or resource from a storage object file. You must first create the storage object file with the retrieve command function. The uncompressed file can be sent to a printer or displayed with the standalone viewer program.

res_diff The res_diff command function can be used to find out if the resources contained in the specified resource group file already exist in OnDemand.

postprocess_cmd

The postprocess_cmd command function can be used to print the postprocessor statements for the specified application on *stdout*. Postprocessor instructions can be specified when you create or update an application with the administrator interface. Refer to administrator online help for more information about how to specify postprocessor instructions for an application.

indexer The indexer command function can be used to print the indexing parameters for the specified application on *stdout*. Indexing parameters can be specified when you create or update an application with the administrator interface. Refer to the administrator interface online help for more information about how to specify indexing parameters for an application.

Parameters

-a application_name

The name of the OnDemand application. You must specify this parameter, if there is more than one application contained in the application group.

-b offset

For the retrieve command function, the offset (in bytes) into the specified storage object where the arsadmin command begins a partial storage object retrieval. The default is the first byte in the storage object.

For the compress command function, the offset (in bytes) from the beginning of the file where the arsadmin command begins data compression. The default is 0 (zero).

For the decompress command function, the offset (in bytes) from the beginning of the file where the arsadmin command begins data decompression. The default is 0 (zero).

-c type

For the compress and decompress command functions, the compression type for source files. Options are 'Z' for LZW16 compression (the default), 'L' for LZW12 compression, 'N' to indicate no compression, and 'D' to disable compression.

-d directory

For the load command function, if you specified the -f parameter to create storage object files, you can use this parameter to specify the directory where the

arsadmin command writes the storage object files. The default is the directory where the *arsadmin* command was invoked.

For the store command function, the name of the directory that contains the storage object file to load. The default is the directory where the *arsadmin* command was invoked.

For the retrieve command function, the directory where the *arsadmin* command writes the file that contains the storage object retrieved from OnDemand. The default is the directory where the *arsadmin* command was invoked.

-e res_ID

For AFP files, this parameter and the **-r** parameter determine how OnDemand processes the resource group file. The default resource ID is 0.

- If you specify a resource ID and a resource file (**-r**), OnDemand loads the resource from the file and overrides the resource specified in the document with the resource ID you specified.
- If you specify a value of 0 (zero) and a resource file (**-r**), OnDemand checks the Server to determine if the resource specified in the document exists. If the resource does exist, OnDemand does not load the resource from the file. Otherwise, OnDemand loads the resource from the file.
- If you specify a value of 0 (zero) and do not specify a resource file (**-r**), OnDemand does not assign a resource ID to the document.
- If you specify a value other than 0 (zero) and do not specify a resource file (**-r**), OnDemand assigns the resource ID you specified to the document.

-f Causes the load command function to write storage objects to file, rather than to storage volumes. By default, OnDemand creates storage objects in memory and stores them in cache storage and on archive media. This parameter can be specified when you want OnDemand to process a report file and create storage objects, but not store the storage objects on storage volumes. You may want to use this option when you are testing the data loading process.

-g appl_group_name

The name of the application group where OnDemand processes the data.

-h library_server

The TCP/IP host name alias, fully-qualified host name, or IP address of the OnDemand library server.

You must specify this flag and the name of the OnDemand library server, when you run the *arsadmin* command on an object server that resides on a different RS/6000 than the library server.

-i index_file

The name of the input index file. Created by the data indexing program. Optionally, for AFP data, the name of a concatenated input file that also contains resource group and report data (also refer to the **-r** and **-s** parameters).

-l length

For the retrieve command function, the number of bytes that the `arsadmin` command retrieves from the specified storage object. The `-l` and `-b offset` parameters provide support for partial storage object retrieval.

For the compress command function, the number of bytes (from the offset) of data to compress. The default is zero bytes.

For the decompress command function, the number of bytes (from the offset) of data to decompress. The default is zero bytes.

-L load_id

A value that OnDemand generates to uniquely identify a report file load. A Load ID contains six parts. For example:

```
1220-1-0-8FAA-9339-9345
```

Where 1220 is the internal application group identifier, 1 is the primary storage node, 0 is the secondary storage node, 8FAA is a counter, 9339 is the first (internal) date in the report file, and 9345 is the last (internal) date in the report file. If you redirected the output of the `ars_load` command or the `load` command function to a log file, check the log file for the Load ID created during load processing.

For the unload command function, the Load ID that you want to delete. When recovering from a system failure or some other problem that occurred during load processing, delete the last (or only) Load ID listed in the log file.

For the `load_db` command function, the Load ID that represents the report file load that generated the index file.

-m location

Determines the location where the storage manager processes the storage objects. Possible locations are cache storage, the primary storage node, and the secondary storage node. The default value is **7** (seven), which causes the storage manager to process the storage objects in all locations defined for the application group. Specify **1** (one) for cache storage, **2** (two) for the primary storage node, or **4** (four) for the secondary storage node. The values are additive, that is, if you specify a value of **3** (three), the storage manager processes the storage objects in cache storage and on the primary storage node.

-n prinid-secnid

The primary and secondary storage node identifiers for the application group. Separate the identifiers with the `-` (dash) character.

OnDemand stores the primary and secondary storage node identifiers in the database, when you load a report file into the application group. OnDemand includes the identifiers in the Load ID.

The values specified with the `-n` and `-g` parameters enable the `arsadmin` command to store data on or retrieve data from a specific OnDemand object server.

-o output_file

For the load command function, optionally create a file that contains the index information that DB2 for AIX uses to update the OnDemand database. You can use this option to create the index data but not load the database, for example, if you want to test the index process or inspect the index data.

For the compress command function, the name of the output (compressed) file.

For the decompress command function, the name of the output (uncompressed) file.

-p password

The password for the OnDemand userid. If you run do not specify this parameter, the arsadmin command prompts you to enter the password at run time.

-r resource_file

The name of the input resource group file. Created by the data indexing program. Optionally, for AFP data, the name of a concatenated input file that also contains index and report data (also refer to the **-i** and **-s** parameters).

-s source_file

For the load command function, the input report file or a concatenated file that contains the report file and index and resource group data (see the **-i** and **-r** parameters).

For the load_db command function, the name of the file that contains the index data to load.

For the compress command function, the name of the input, uncompressed file.

For the decompress command function, the name of the storage object file that contains the item to be decompressed.

-u userid

An OnDemand user with administrator authority for the application group. If you do not specify this parameter, the arsadmin command prompts you to enter the userid at run time.

For the unload command function, the userid that you specify must have delete document permission for the application group.

-y delimiter

The column delimiter. Optionally specify the character that separates the columns (fields) in the database rows contained in the index file passed to DB2 for AIX. The default delimiter is the “|” (pipe symbol).

OnDemand uses the delimiter when you specify the **-o** flag, to write the database input records to file.

filename_n

For the store command function, the names of one or more storage object files to load.

For the retrieve command function, the names of one or more storage objects to retrieve. Also the names of the files that the arsadmin command creates in the **-d** directory.

For the `res_diff` command function, the names of one or more resource group files to process.

table_name

For the import command function, the name of the migrated index table to import into the database.

Examples

1. The following example shows how to invoke the load command function to load report file data contained in the `credit.out` file and index data contained in the `credit.ind` file into the Credit application group. The OnDemand user is `admin`. There is no password assigned to the user.

```
/usr/lpp/ars/bin/arsadmin load -g 'Credit' -u admin -p ""  
-i credit.ind -s credit.out -r credit.res
```

2. The following example shows how to invoke the load command function to load a concatenated input file that contains the report file, index data, and resources.

```
/usr/lpp/ars/bin/arsadmin load -g 'Credit' -u admin -p ""  
-i credit.cat -s credit.cat -r credit.cat
```

3. The following example shows how to invoke the unload command function to delete a Load ID from OnDemand.

```
/usr/lpp/ars/bin/arsadmin unload -g 'Credit'  
-L 1220-1-0-1FAA-9339-9345 -u admin -p ""
```

4. The following example shows how to use the compress command function to estimate the storage required for a report file that contains logical items, such as bills, invoices, or statements. The `arsadmin` command compresses the first 200 KB of data in the report file.

```
/usr/lpp/ars/bin/arsadmin compress -s report_file  
-o output_file -l 200000
```

5. The following example shows how to use the compress command function to estimate the storage required for report files that contain transaction data, such as a general ledger. The input file is a portion of a larger report file.


```
/usr/lpp/ars/bin/arsadmin compress -s report_file_segment  
-o output_file
```

6. The following example shows how to use the import command function to import a migrated index table from archive media into the database. The name of the application group and the index table to import can be obtained from the message written to the system log.

```
/usr/lpp/ars/bin/arsadmin import -g 'Credit'  
-u admin -p "" SL27
```

Files

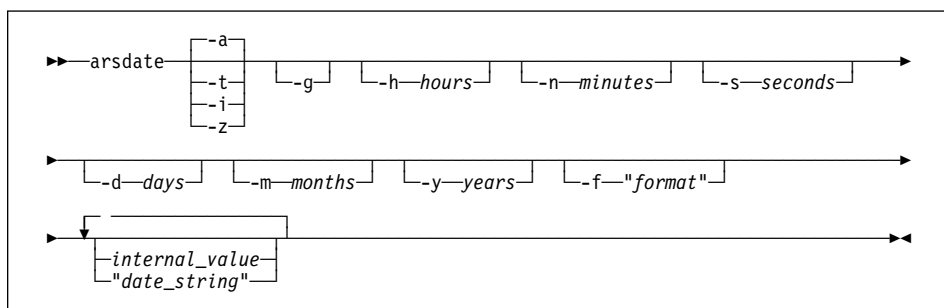
/usr/lpp/ars/bin/arsadmin	The executable program.
----------------------------------	-------------------------

Chapter 20. ardate command reference

Purpose

Display the OnDemand internal date value for a given date string or display the date string for a given OnDemand internal date value.

Syntax



Description

The `ardate` command can be used to display a date string using the internal date value that OnDemand stores in the database. You can also use the `ardate` command to display the internal value for a given date string.

Parameters

- a** Display the OnDemand internal date format for the specified date string.
- t** Display the OnDemand internal time format for the specified time string.
- i** Display the OnDemand internal date and time format for the specified date and time string.
- z** Display the OnDemand internal date and time format for the specified date and time string, using the time zone of the server.
- g** Display the current date.
- d days** Add the specified number of days to the specified internal value.
- h hours** Add the specified number of hours to the specified internal value.
- m months** Add the specified number of months to the specified internal value.
- n minutes** Add the specified number of minutes to the specified internal value.
- s seconds** Add the specified number of seconds to the specified internal value.

- y years** Add the specified number of years to the specified internal value.
- f "format"** Determines the format of the date and time string that OnDemand displays.

The following table lists some of the formats supported by OnDemand. If the format that you require is not listed, you can create your own, using values, separators, and the percent character.

The format consists of a set of values, for example, **%m**, and separators, such as the blank character.

Table 3. Date and Time Format Specifications

Date Format Specifier	Date Format	Example
%m/%d/%y	mm/dd/yy	01/31/95
%d/%m/%y	dd/mm/yy	31/01/95
%f/%e/%y	m/d/yy	1/31/95
%e/%f/%y	d/m/yy	31/1/95
%m-%d-%y	mm-dd-yy	01-31-95
%d-%m-%y	dd-mm-yy	31-01-95
%m%d%y	mmddy	013195
%m%d%Y	mmddyyyy	01311995
%y%m%d	yymmdd	950131
%Y%m%d	yyyymmdd	19950131
%b %e, %Y	Mth d, yyyy	Jan 31, 1995
%B %e, %Y	Month d, yyyy	January 31, 1995
%m/%d/%y %H:%M	mm/dd/yy hh:mm	01/31/95 10:50
%H:%M	hh:mm	10:50
%T	hh:mm:ss	10:50:59

internal_value

The internal date value from the OnDemand database. Enter one or more internal date values when you want to display formatted date strings.

"date_string"

The date string. Enter one or more date strings when you want to display internal date values.

Examples

1. To obtain the date in the default format for a given internal value, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate 9404
```

The *arsdate* command displays:

```
9404 -> 09/30/95
```

2. To obtain the dates in the default format for a given set of internal values, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate 9404 9374 9343
```

The *arsdate* command displays:

```
9404 -> 09/30/95  
9374 -> 08/31/95  
9343 -> 07/31/95
```

3. To obtain the date using a specific format for a given internal value, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate -f "%B %e, %Y" 9404
```

The *arsdate* command displays:

```
9404 -> September 30, 1995
```

4. To obtain the current date, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate -g
```

The *arsdate* command displays:

```
Today: 07/24/95
```

5. To obtain the internal date for July 24, 1995, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate -a 7/24/95
```

The *arsdate* command displays:

```
7/24/95 -> 9336
```

6. To obtain the internal dates for a set of given dates, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate -a 7/31/95 8/31/95 9/30/95
```

The *arsdate* command displays:

```
7/31/95 -> 9343  
8/31/95 -> 9374  
9/30/95 -> 9404
```

Files

/usr/lpp/ars/bin/arsdate

The executable program.

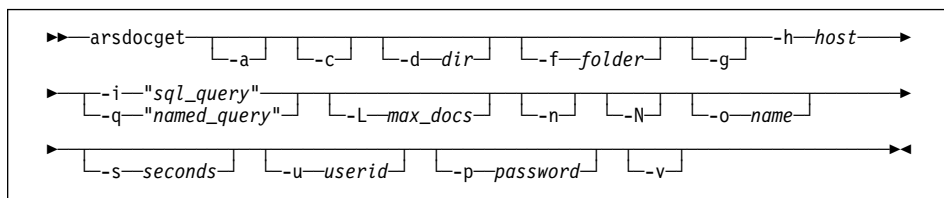
Chapter 21. arsdocget command reference

Purpose

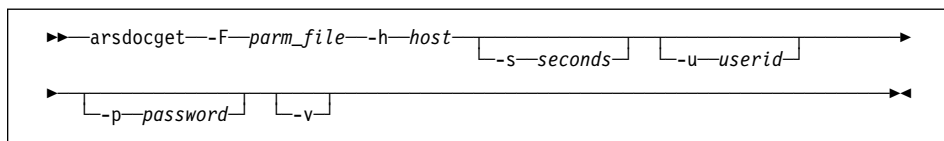
The arsdocget command is a batch program that can be used to retrieve documents from OnDemand.

Syntax

You can identify the folder, query, output file, and other values, and options on the command line or you can specify the name of a parameter file that contains this information. The following diagram shows the syntax of the arsdocget command when you do not use the parameter file option.



When you use the parameter file option, you must not specify the -a, -c, -d, -f, -g, -i, -L, -n, -N, -o, or -q flags on the command line. The following diagram shows the syntax of the arsdocget command when you use the parameter file option.



Description

You can invoke the arsdocget command from the AIX prompt or from a script file to retrieve documents and their associated resources from OnDemand. The arsdocget command can also generate and save generic index data for the documents that match the query. You identify the name of an OnDemand library server, the name of the folder that you want to search, and specify one or more queries to run. To query the database, you can enter an SQL query that contains clauses, database field names, index values, and operators or you can enter the name of a public named query. If you do not identify the name of an OnDemand userid authorized to run the query, the arsdocget command prompts you to enter the userid and password at run time. By default, the arsdocget command sends a copy of the documents that match the SQL query to *stdout*. You typically redirect the output to a printer or another process. You can also write the output to a file.

Parameters

- a** For AFP documents, specify this parameter to include resources in the output data stream. You typically use this parameter when redirecting the output of the `arsdocget` command to a print device or another process. If you specify the **-g** flag, do not specify the **-a** flag.

If more than one version of the resource group exists for an item that matches the query, the `arsdocget` command creates a separate output data stream for each version of the resource group. For example, if two versions of the resource group exist for an item that matches the query, the `arsdocget` command creates two output data streams. Each output data stream contains the document and one version of the resource group.

- c** Specify this parameter to concatenate all of the items that match the query in one output file. By default, the `arsdocget` command writes each item that matches the query to a separate output file.

If you specify the **-c** flag, you must specify the **-o** flag and name the output file. However, the `arsdocget` command ignores the **-c** flag if you specify the **-o** flag and identify database field names. That is, when you generate file names using database field names, the `arsdocget` command automatically stores each item that matches the query in a separate output file. In addition, the `arsdocget` command ignores the **-c** flag and creates separate output files when any of the following conditions occur:

- If more than one application group is referenced by the folder. The `arsdocget` command creates one output file for each application group that contains items that match the query.
- If more than one application is contained in an application group. The `arsdocget` command creates one output file for each application that contains items that match the query.

For example, if a folder references two application groups, the following specification:

```
-o student -c
```

Can result in file names such as:

```
student.516.out  
student.517.out
```

Where 516 and 517 are application or application group identifiers. One file is created for each application or application group. Each file contains all of the items that match the query for that particular application or application group.

-d directory

The name of the directory where arsdocget writes the output files. The directory must exist before the arsdocget command attempts to write the output files. Directory names are case sensitive.

-f folder

The name of the folder to query. The folder name must be specified exactly as it appears in OnDemand. The case of the folder name is significant. For example, to query the OnDemand system log folder, you must enter:

```
-f "System Log"
```

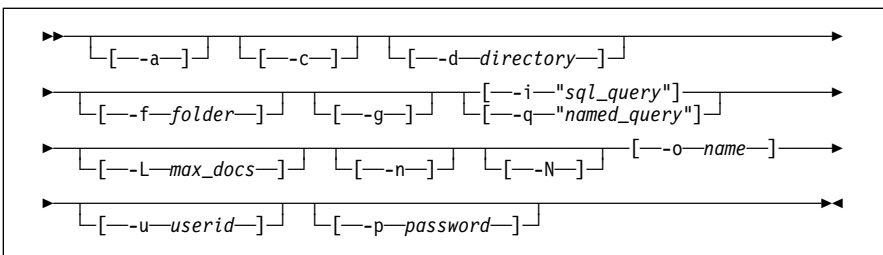
The folder can reference one or more application groups. If there are multiple application groups referenced by the folder, all of the application groups must have common database field names. For example, if the application groups include a date field, the name of the date field in all of the application groups must be the same.

If you do not specify this parameter and value, the arsdocget command prompts you for the folder name at run time.

-F parm_file

Determines the name of the file that contains the name of the folder, the queries, and other document and file parameters, values, and options. Specify this option when you want to perform multiple queries.

Each line in the parameter file represents a different query and can contain the following options:



Flags and values specified in the parameter file must be enclosed in brackets. You must specify the **-i** or **-q** flag. The items that match the query must be written to output files. That is, you must specify the **-o** flag.

-g Specify this parameter to generate generic index data for the items that match the query. The arsdocget command creates one generic index file for each application group queried. Refer to the *Indexing Reference* for details about the format of the generic index file.

When you specify the **-g** flag, you must specify the **-c** and **-N** flags and you cannot specify database field names with the **-o** flag.

The `arsdocget` command uses the following convention to name output files generated with the `-g` flag:

`-o.res_id.appl_group.appl.type`

Where:

`-o` is the value specified with the `-o` flag.

`res_id` is the resource group identifier. A resource group identifier always starts at 1 (one) and is incremented by one for each version of the resource group.

`appl_group` is the name of the application group.

`appl` is the name of the application.

`type` is the file type:

- **out** identifies a document file. The `arsdocget` command creates one document file for each application group.
- **ind** identifies a generic index file. The `arsdocget` command creates one generic index file for each application group. For AFP documents, the `arsdocget` command also creates one generic index file for each version of the resource group per application group.
- **res** identifies a resource file. For AFP documents, the `arsdocget` command creates one resource file for each version of the resource group per application group.

-h host

The host name or IP address of the OnDemand library server that contains the folder that you want to query and the items that you want to retrieve. This is a required parameter.

-i "sql_query"

A valid SQL query, that includes the names of one or more application group database fields, index values, and operators. OnDemand does not validate the string that you enter. Please refer to the *DB2 for AIX SQL Reference* for an overview of SQL concepts and details about how to construct a query.

If you specify the `-i` flag to query OnDemand, you cannot specify the `-q` flag.

To construct a query with a database field of type date, you must use the OnDemand internal format of the date. That is, the number of days since January 1, 1970. You can use the `arsdate` command to display the internal format for a given date. For example, to obtain the internal date for July 21, 1995, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate -a 7/21/95
```

The `arsdate` command displays:

```
7/21/95 -> 9333
```

Enter 9333 as the index value for the date database field.

-L max_docs

Determines the maximum number of items retrieved from OnDemand, regardless of the number of items that match the query. OnDemand retrieves items in the order that they were loaded in the database.

-n Specify this parameter to retrieve items one at a time from the database. By default, the arsdocget command uses bulk retrieval query processing for high-speed retrieval of items from the database.

-N When querying a folder that references more than one application group or a folder that references an application group that contains more than one application, specify this parameter to add the resource identifier, application group name, and application name to the output file name. When you specify the **-N** flag, you must specify the **-c** flag. If you specify the **-g** flag to generate generic index data, you must specify the **-N** flag.

If the folder references more than one application group or an application group contains more than one application and you do not specify the **-N** flag, the arsdocget command adds the application group or application identifier to the output file name. For example, the following specification:

```
-o student -c
```

Can result in output file names such as:

```
student.516.out  
student.517.out
```

Where 516 and 517 are application group or application identifiers. However, when you specify the **-N** flag, the arsdocget command uses the resource identifier, application group name, and application name to name the output file. For example, the following specification:

```
-o student -c -N
```

Can result in output file names such as:

```
student.1.BILLS.1995.out  
student.1.BILLS.1996.out
```

Where 1 is the resource identifier, BILLS is the application group name, and 1995 and 1996 are application names.

When generating generic index data for AFP documents, the `arsdocget` command creates one or more index files based on the number of resource groups processed. One index file is created for each resource group processed per application group. For example, if two resource groups are processed for an application group, the `arsdocget` command can create files such as:

```
student.1.BILLS.1995.ind
student.1.BILLS.1995.out
student.1.BILLS.1995.res
student.2.BILLS.1995.ind
student.2.BILLS.1995.res
```

-o name

Directs the `arsdocget` command to write documents to one or more files and identifies the user-defined string used to generate unique file names. For example, the following specification:

```
-o student -c
```

Can result in the following output file name:

```
student
```

The user-defined string can contain one or more characters, except the following:

\	Backward slash	/	Forward slash
"	Double quote	~	Tilde
'	Single quote	*	Asterisk
;	Semicolon	?	Question mark
!	Exclamation point	\$	US dollar
{	Left brace	}	Right brace
[Left Bracket]	Right Bracket
#	Number sign	@	At sign
&	Ampersand		Vertical bar
	Blank character	-	Dash

In addition, if you use database field names to generate file names, the ((left parenthesis) and) (right parenthesis) characters cannot be used.

The file name generated by the `arsdocget` command cannot exceed 255 characters, including information automatically appended to the string that you specify. Refer to the `-c`, `-g`, and `-N` flags for details.

You can concatenate one or more of the database field names that you specify with the **-i** flag to generate a unique file name. For example, the following specification:

```
-o "(student).(type)"  
-i "WHERE type='T' AND student='001200340056'"
```

Can result in the file name:

```
001200340056.transcript
```

When you use database field names to generate a unique file name:

- OnDemand verifies that the field names that you specify are valid for the application group(s) referenced by the folder specified with the **-f** flag.
- If the field name that you specify is a date field, the output format of the date is determined by the Defaults Fmt folder field setting.
- The field names that you specify must be delimited with parenthesis.
- You can specify the fields in any order. The order that you specify determines the file name that arsdocget generates.
- You cannot use a field name to represent a directory name. For example:

```
-o "/(field_1)/(field_2)"
```

Is not valid.

- The entire file name string must be enclosed in double quotes.
- You cannot specify the **-c** flag to concatenate items in one output file. Each item that matches the query is stored in a separate output file.

If more than one item matches a query and you do not generate a unique file name using database field names, concatenate items in a single file with the **-c** flag, or specify the **-g** flag, the arsdocget command generates a unique file name for each item that matches the query by adding a *.n* extension to the file name. Where *n* is the number of the item that matched the query. For example, if you specify:

```
-o statements
```

And two items match the query, the arsdocget command creates the following files:

```
statements.1
statements.2
```

You must specify the **-o** flag when you specify the **-c** flag.

—q "named_query"

The name of a public named query for the folder named with the **-f** flag. A named query is a set of search criteria previously saved on the library server that can be recalled by name to search a folder. A named query is typically defined to search a folder for a specific set of documents.

If you specify the **-q** flag to query OnDemand, you cannot specify the **-i** flag.

-s seconds

Optionally determines the number of seconds that the `arsdocget` command waits between query requests, when you specify more than one query with the **-F** flag. If you do not specify this option, OnDemand does not wait between query requests. That is, the default is 0 (zero) seconds.

-u userid

The name of an OnDemand user. The `arsdocget` command verifies that the user name that you specify is a valid OnDemand user for the server that you name with the **-h** flag. The `arsdocget` command verifies that the user name that you specify is authorized to open the folder that you name with the **-f** flag. If you do not specify the **-u** flag, the `arsdocget` command prompts you for the user name and password at run time.

-p password

The password of the OnDemand user that you named with the **-u** flag. If there is no password assigned to the user that you specify, use double quotes to show a null password. That is, specify **-p ""**. If you do not specify the **-u** flag, the `arsdocget` command prompts you to enter the password at run time. If there is no password assigned to the user that you specify, press the Enter key when prompted.

-v Enables verbose mode. When you specify this flag, the `arsdocget` command displays additional messages. By default, the `arsdocget` command only displays error messages.

Examples

1. The following example shows how to invoke the `arsdocget` command using the command line parameters to retrieve an item from OnDemand and save a copy of the item in a file in the current directory. The `arsdocget` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsdocget -h rockies -f "student information"
-o student -c
-i "WHERE student='001200340056'"
```

The `arsdocget` command concatenates all items that match the query in the following output file:

```
student
```

2. The following example shows how to invoke the `arsdocget` command and specify the name of a parameter file in the current directory that contains a set of queries. The results of each query are saved in separate files in the current directory. The `arsdocget` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsdocget -h rockies -F parmfile
```

The parameter file contains two queries (note the use of the AIX line continuation character):

```
[-f "student information"] \  
[-i "WHERE student='123420010056' AND type='B' OR type='G' OR type='T'"] \  
[-o "(student).(type)"] \  
[-f "student information"] \  
[-i "WHERE student='123450011917' AND type='B' OR type='G' OR type='T'"] \  
[-o "(student).(type)"]
```

Assuming items exist in the database for all types, the `arsdocget` command creates the following output files:

```
123420010056.bills  
123420010056.grades  
123420010056.transcripts  
123450011917.bills  
123450011917.grades  
123450011917.transcripts
```

3. The following example shows how to invoke the `arsdocget` command using the command line parameters to retrieve an item from OnDemand and send a copy of the item to a printer using the `enq` command. The `arsdocget` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsdocget -h rockies -f "student information"
-a -i "WHERE type='T' AND student='001200340056'"
| enq -P G4028
```

4. The following example shows how to invoke the `arsdocget` command using the command line parameters to retrieve items from OnDemand, write the documents to a file, and generate and save generic index data for the items that match the query. The name of a public named query is identified. The `arsdocget` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsdocget -h rockies -f "student information"
-a -c -g -o student -q "3rd yr students GPA>3.5" -N -v
```

The number of output files generated is a factor of the number of application groups queried, the applications contained in the application groups, whether the data is AFP, and if so, the versions of resource groups in each application group. At a minimum, for AFP data with one version of the resource group and one application group, the `arsdocget` command generates three output files. For example:

```
student.1.student information.TRANSSCRIPTS.ind
student.1.student information.TRANSSCRIPTS.out
student.1.student information.TRANSSCRIPTS.res
```

For AFP data, if there is more than one version of the resource group per application group, the `arsdocget` command can generate additional output files. For example:

```
student.1.student information.TRANSSCRIPTS.ind
student.1.student information.TRANSSCRIPTS.out
student.1.student information.TRANSSCRIPTS.res
student.2.student information.TRANSSCRIPTS.ind
student.2.student information.TRANSSCRIPTS.res
```

If the application group contains more than one application, the `arsdocget` command can generate additional output files. For example:


```
student.1.student information.BILLS.ind  
student.1.student information.BILLS.out  
student.1.student information.BILLS.res  
student.1.student information.GRADES.ind  
student.1.student information.GRADES.out  
student.1.student information.GRADES.res
```

Files

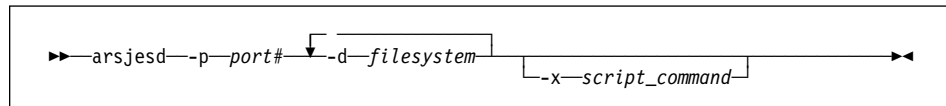
<code>/usr/lpp/ars/bin/arsdocget</code>	The executable program.
---	-------------------------

Chapter 22. arsjesd command reference

Purpose

The arsjesd command monitors a TCP/IP port for datasets transmitted by MVS Download and stores data in file systems on the RS/6000.

Syntax



Description

The arsjesd command is the AIX component of the MVS Download feature. MVS Download is an optional feature of PSF for MVS that can be used to automatically transmit application program output datasets at high speed, from the JES spool to file systems on the RS/6000. MVS Download also transmits values you specify in JCL parameters and can optionally transmit additional job information. You can use these values to specify the name of the application and application group where OnDemand loads the data. The MVS Download feature is made up of a pair of cooperating components, an MVS Download writer that transmits the dataset and an AIX program that receives and stores the data on the RS/6000. Refer to the *PSF for MVS: MVS Download Guide* for details about MVS Download, including how to configure and operate MVS Download on the MVS system, a list of the JCL parameters transmitted with a file, and information about the user exit you can use to provide additional job information to the arsjesd command.

The arsjesd command monitors a TCP/IP port for activity from an MVS system. Multiple instances of the arsjesd command can be started, each monitoring a specific port for data transmitted from a different MVS Download writer. An MVS Download writer and its corresponding arsjesd instance must specify the same port number. If you start multiple instances of the arsjesd command, make sure that you specify difference file systems for each instance. Overlapping file systems can adversely affect performance.

The arsjesd command stores datasets received from an MVS system in file systems on the RS/6000. You can specify up to ten file systems for the arsjesd command to store data transmitted from an MVS system. The arsjesd command always stores data beginning with the first file system specified. If there is not enough space available to store the dataset being transmitted on the first file system specified, the arsjesd command switches to the second file system specified. If there is not enough space available to store the dataset being transmitted on the last file system specified, the arsjesd command switches to the first file system specified.

Parameters

- p port#** The TCP/IP port number that the arsjesd command monitors for data from an MVS system. The valid range of port numbers is 5001 through 64000. However, avoid using port numbers 5001, 8251, and 8253. The MVS Download writer and the arsjesd command must specify the same port number. Refer to the *PSF for MVS: MVS Download Guide* for information about configuring MVS Download on the MVS system.
- d filesystem** The name of the file system where the arsjesd command stores data transmitted from an MVS system. You can specify up to 10 file system parameters, separated by intervening blanks. Planning a Storage Subsystem in the *Introduction and Planning Guide* describes a recommended naming convention for file systems that support MVS Download.
- x script_command** The name of a shell script file that the arsjesd command invokes to process a downloaded file and the JCL parameters after receiving and storing a file on the RS/6000. You can specify the full path name of the shell script file. If you do not specify the full path name, AIX searches the current directory for the script file. If the file is not found in the current directory, AIX continues to search for the file using the directories defined in the PATH environment variable.
- If you do not specify this parameter along with the name of a script file, OnDemand can process the downloaded file using other means. For example, you can invoke the `ars_load` command from the AIX prompt or you can set up the `ars_load` command to run as an AIX daemon. Refer to Chapter 18, “ars_load command reference” on page 111 for information about processing data with the `ars_load` command. The *Installation and Configuration Guide* provides instructions about how to automate the process of receiving files from an MVS system and processing data on the RS/6000.
- Chapter 5, “Data download exits and processing” on page 29 provides information about using the arsjesd command to invoke a shell script to process a downloaded file and the JCL parameters.

Examples

1. The following example shows how to start the arsjesd command to monitor TCP/IP port number 6001 and place datasets downloaded from the MVS system in the /arsacif/acif1 file system.

```
/usr/lpp/ars/bin/arsjesd -p 6001 -d /arsacif/acif1
```

2. The following example shows how to start the `arsjesd` command to monitor TCP/IP port number 6002 and place datasets downloaded from the MVS system in the `/arsacif/acif1` file system. If there is not enough space available in the `/arsacif/acif1` file system to store the dataset being transmitted, the `arsjesd` command stores the dataset in the `/arsacif/acif2` file system.

```
/usr/lpp/ars/bin/arsjesd -p 6002
-d /arsacif/acif1 -d /arsacif/acif2
```

3. The following example shows how to start the `arsjesd` command to monitor TCP/IP port number 6002 and place datasets downloaded from the MVS system in the `/arsacif/acif1` file system. After transmitting the file to the RS/6000, the `arsjesd` command invokes the `/usr/lpp/ars/bin/mvs_download` shell script. The script file can perform several functions, such as determining the name of the application group to load using the value of one of the JCL parameters passed to the `arsjesd` command.

```
/usr/lpp/ars/bin/arsjesd
-p 6002 -d /arsacif/acif1
-x /usr/lpp/ars/bin/mvs_download
```

Notes

MVS Download uses the following file naming convention:

```
HLQ.JOBNAME.DATASET.FORMS.YYDD.HHMMSS.ARD
```

- The `FORMS` part of the filename can be used to identify the application group to load. Optionally, you can use the `-g` flag to identify the application group.
- If the application group to load contains more than one application (source of data), the `DATASET` part of the filename can be used to identify the application within the application group. However, you must modify the `ars_load` shell script to enable this function. Otherwise, you must either specify the name of the application when you invoke the `ars_load` command (with the `-a` flag) or process the JCL parameters transmitted with the downloaded file to identify the application before you invoke the `ars_load` command. Please refer to Chapter 5, “Data download exits and processing” on page 29 for more information about processing files and JCL parameters transmitted by MVS Download.

Please refer to the *PSF for MVS: MVS Download Guide* for more information about MVS Download, including details about user exits, JCL parameters that support

transmitting files from an MVS system to an OnDemand system, and how to install, configure, and operate MVS Download on the MVS system.

Files

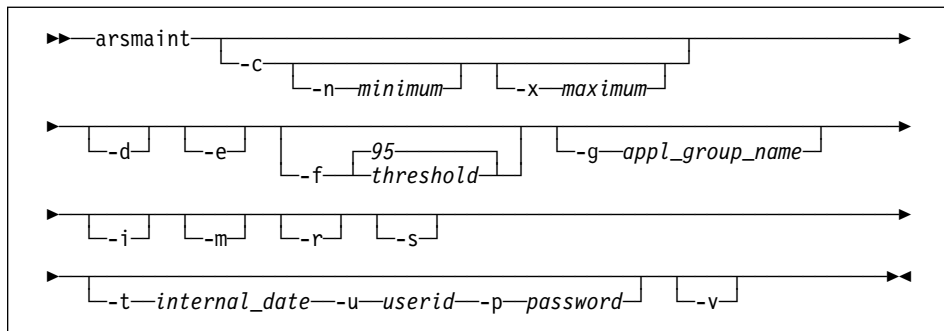
<code>/usr/lpp/ars/bin/arsjesd</code>	The executable program.
---------------------------------------	-------------------------

Chapter 23. arsmaint command reference

Purpose

Maintain application group data in the OnDemand database and cache file systems. You typically run the arsmaint command every day, to migrate index data to archive media, delete index data from the database, migrate files from cache file systems to archive media, and delete files from cache storage.

Syntax



Description

The `arsmaint` command maintains application group data in the OnDemand database and cache storage based on the storage management values that you provide for application groups. The `arsmaint` command uses values that you specify for the life of data and indexes, length of time to cache data on magnetic, length of time before copying cache to archive media, length of time before migrating indexes to archive media, and length of time to maintain imported migrated indexes. The `arsmaint` command uses the expiration type to determine how to delete index data from an application group. The `arsmaint` command can expire a segment (table) of application group data at a time, a report file load at a time, or individual items.

The `-d`, `-e`, and `-i` flags cause the `arsmaint` command to migrate and expire index data from the database.

The `-c` and `-m` flags cause the `arsmaint` command to migrate and expire files from cache storage.

Note: The business and operational requirements of your organization and the configuration of your system determine when you can run and the time available for maintenance, such as reorganizing and backing up the database. Most database maintenance should take place when there is little or no other activity (especially loading data or administering the system). Some maintenance tasks

must run when the system is idle. You should carefully schedule maintenance processes, to avoid problems, such as locking database tables. For example, we recommend that you do not run the `ars_load` command at the same time that you run database statistics (for example, running the `arsmaint` command with the `-r` option or the `ars_db` command with the `-s` option) or perform maintenance on the database (for example, running `ars_db` with the `-m` option). There may be other processes that conflict; for example, you cannot load data or maintain the database while an offline backup of the database is in progress. The time of day and frequency with which you run the maintenance programs and the processing options you specify will vary, based on your environment. For example, if you do not load data every day or you add few rows to the database when you do load data, you can schedule database maintenance less often, perhaps once a month. Conversely, if you load a high volume of data every day, we recommend that you maintain the database after the load processing completes. In addition, if your system has plenty of space available to store index data and reports, you may be able to maintain the system less often. The examples in this chapter demonstrate how to set up the database maintenance processes using typical options: the use of cache storage, migrating data to archive media, and maintaining the database and cache storage every day. Use these examples as a guideline. The exact syntax of the commands and when you can run them on your system will vary.

Parameters

- c** Expire files from cache storage. We recommend that you schedule the `arsmaint` command to run every day using this parameter.
- d** Expire indexes from the OnDemand database. We recommend that you schedule the `arsmaint` command to run every day using this parameter.
- e** Migrate index data to archive media. We recommend that you schedule the `arsmaint` command to run every day using this parameter.

Migration is the process of moving tables of index data from the database to archive media. This process optimizes database storage space while allowing you to maintain index data for a very long time. You typically migrate index data after users no longer need to access the reports, but, for legal or other requirements, you still need to maintain the data for some number of years or months.

You must configure your application groups for migration. OnDemand uses the length of time before migrating indexes value to determine when to migrate index data. After maintaining the imported index data for the number of days specified in the length of time to keep imported migrated indexes field, OnDemand expires the data from the database. Refer to the administrator interface online help for details.

Before OnDemand can migrate index data to archive media, you must assign the System Migration application group to a storage set that writes to archive media. Refer to the *Installation and Configuration Guide* for details.

-f threshold

Send an alert message when a cache file system is filled to the percentage specified. The default value is 95.

When a cache file system exceeds the threshold, it usually means that the `arsmaint` command cannot delete enough items from the file system to drop below the threshold. You may need to adjust the data migration and expiration values of the application groups using the file system. Otherwise, increase the size of the file system or add a new cache file system.

The `arsmaint` command sends the alert message to the console, the system log, and the system log user exit.

-g appl_group_name

The name of the application group to process. By default, the `arsmaint` command maintains the database or cache storage for all of the application groups defined to OnDemand. Specify this parameter and the name of an application group when you want to maintain the database or cache storage for a specific application group. The `-g` parameter is valid only with the `-c`, `-d`, `-e`, `-i`, `-m`, and `-r` parameters.

- i** Expire imported index data from the database. We recommend that you schedule the `arsmaint` command to run every day using this parameter.

An administrator can import index data previously migrated to archive media back into the database to satisfy a query. After maintaining the imported index data for the number of days specified in the length of time to keep imported migrated indexes field, OnDemand expires the data from the database.

- m** Migrate files from cache storage to archive media. We recommend that you schedule the `arsmaint` command to run every day using this parameter.

-n minimum

The low expiration threshold percentage, per cache file system. This value determines when the `arsmaint` command stops expiring files from cache file systems (see the `-c` and `-x` flags).

The default value is 80 (percent), which keeps a cache file system fairly full to provide good retrieval hit rates, while leaving free space for report file loads. However, we recommend that you periodically check the amount of available space in your cache file systems, especially before scheduling very large report file loads. You may need to increase the size of one or more cache file systems, add a new cache file system, or manipulate the data migration and cache management values to provide sufficient free space for large report file loads.

-p password

The password for the OnDemand userid identified with the `-u` flag. This parameter is required when you specify the `-t` flag.

- r** Run database statistics. Causes the database manager to optimize index data. Makes access to information as efficient as possible. We recommend that you schedule the `arsmaint` command to run every day using this parameter.

-s Print a report that contains status information about the cache file systems. The report includes information such as the name of the file systems, the amount of space allocated to each file system, and the amount of free space for each file system. The `arsmaint` command places the report in the system log. We recommend that you schedule the `arsmaint` command to run every day using this parameter.

-t internal_date

Specify this parameter when you want to use the `arsmaint` command to process the database and cache storage using a date other than the current system date (the default). For example, you can specify a future date to cause the `arsmaint` command to process data and files before the actual expiration/migration period. The value that you specify must be a valid OnDemand internal date value. You can use the `arsdate` command to display the internal date value for a given calendar date. When you specify this option, you must specify the **-u** and **-p** flags. The `arsmaint` command processes data and files belonging to the application groups, where the user (**-u**) has the delete document permission enabled.

Note

Use this option with extreme caution. We recommend that you contact the IBM support center before running the `arsmaint` command with this option.

-u userid

An OnDemand user with administrator permission or, if you specify the **-g** flag, permission to delete documents from the application group. This parameter is required when you specify the **-t** flag.

-x maximum

The high expiration threshold percentage, per cache file system. This value determines when the `arsmaint` command begins expiring files from cache file systems (see the **-c** flag). When a cache file system reaches this threshold, the `arsmaint` command searches the file system and expires every file that meets the data migration and caching values. The `arsmaint` command stops expiring files when the amount of used space in a cache file system drops below the low expiration threshold (see the **-n** flag).

The default value is 80 (percent), which keeps a cache file system fairly full to provide good retrieval hit rates, while leaving free space for report file loads. However, we recommend that you periodically check the amount of space available in your cache file systems, especially before scheduling very large report file loads. You may need to increase the size of one or more cache file systems, add a new cache file system, or manipulate data migration and cache management values, to provide sufficient free space for large report file loads.

-v Validates cache storage. When you specify this option, the `arsmaint` command inspects all cache file systems to ensure that they are correctly linked, with the proper file permissions.

Examples

1. The following example shows how to run the `arsmaint` command with options to migrate and expire files from cache storage, migrate and expire indexes from the database, optimize database index data, print statistics, and validate cache storage.

These options are typical of those used to maintain a library server and object server with archive media on the same RS/6000.

```
/usr/lpp/ars/bin/arsmaint -cdeimrsv
```

2. The following example shows how to run the `arsmaint` command with options to migrate and expire files from cache storage, print statistics, and validate cache storage.

These options are typical of those used to maintain an object server with archive media on a different RS/6000 than the library server.

```
/usr/lpp/ars/bin/arsmaint -cmsv
```

Notes

Please refer to the administrator interface online help for details about defining storage management information for application groups.

OnDemand server programs can remain active while the `arsmaint` command processes. However, we recommend that you schedule the `arsmaint` command at a time of little or no OnDemand client program activity.

We recommend that you do not run the `ars_load` command at the same time that you run the `arsmaint` command with the `-r` option.

We recommend that you create a backup image of the OnDemand database and the ADSM database before you run the `arsmaint` command.

Files

<code>/usr/lpp/ars/bin/arsmaint</code>	The executable program.
--	-------------------------

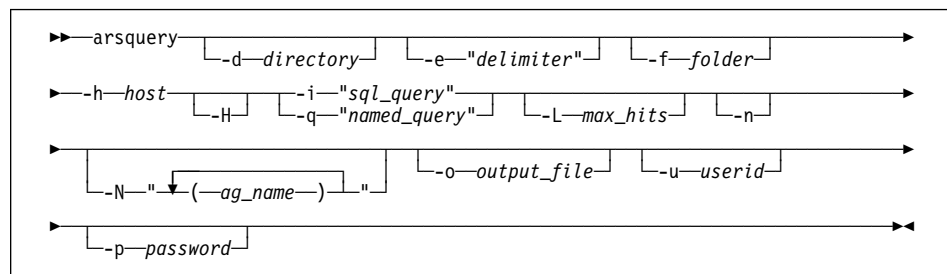
Chapter 24. arsqquery command reference

Purpose

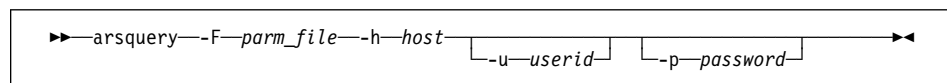
The arsqquery command is a batch program that can be used to construct a list of items that match a query.

Syntax

You can specify the folder, SQL query, output file and directory parameters, values, and options on the command line or you can specify the name of a parameter file that contains this information. The following shows the syntax of the arsqquery command when you do not use the parameter file option.



When you use the parameter file option, you must not specify the -d, -e, -f, -H, -i, -L, -n, -N, -o, or -q flags on the command line. The following shows the syntax of the arsqquery command when you use the parameter file option.



Description

You can invoke the arsqquery command from the AIX prompt or from a script file to construct a list of items that match a query. You specify the name of an OnDemand server, the name of the folder that you want OnDemand to search, and an SQL query that contains clauses, database field names, index values, and operators. If you do not specify an OnDemand user and the password for the user, the arsqquery command prompts you to enter the information at run time. By default, the arsqquery command sends the list of items that match the query to *stdout*. Typically you redirect the output to a file using the **-o** flag.

Parameters

-d directory

The name of the directory where arsquery writes the output file. Do not specify this option from the command line when you specify the **-F parm_file** option.

-e "delimiter"

Determines the character that separates the database field values in the output file. By default, OnDemand separates values in the output file with a comma. Do not specify this option from the command line when you specify the **-F parm_file** option.

-f folder

Determines the name of the folder that you want to query. The folder name must be specified exactly as it appears in OnDemand. For example, to query the OnDemand system log folder, you must enter:

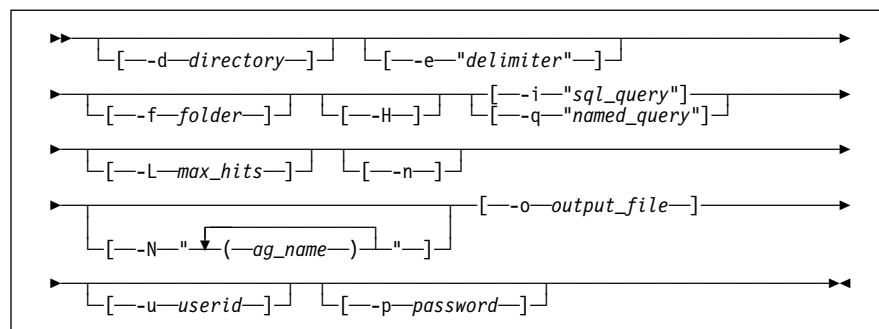
```
-f "System Log"
```

That is, the case of the folder name is significant.

The folder can contain one or more application groups. If you do not specify this parameter and value, the arsquery command prompts you for the folder name at run time. Do not specify this option on the command line when you specify the **-F parm_file** option.

-F parm_file

Determines the name of the file that contains the name of the folder, SQL query string, and other document and file parameters, values, and options. Specify this option when you want to enter multiple query requests. Each line in the parameter file represents a different query and can contain the following options:



Parameters and values must be enclosed in brackets. You must specify either the **-i** or the **-q** flag. The list of items that match the query must be written to a file. That is, you must specify the **-o** flag.

-h host

The host name or IP address of the OnDemand server that contains the folder that you want to query. This is a required parameter.

- H** Specify this parameter to generate a header record in the output file. The default header record contains the folder field names, delimited with commas. If you select database field names with the **-N** parameter, the folder field names that correspond to the selected database field names appear in the header record. You can specify a field delimiter with the **-e** parameter, so that the field names in the header record are delimited with a character other than the comma. Do not specify this parameter from the command line when you specify the **-F parm_file** option.

-i "sql_query"

A valid SQL query, that includes the names of one or more application group database fields, index values, and operators. OnDemand does not validate the string that you enter. Please refer to the *DB2 for AIX SQL Reference* for an overview of SQL concepts and details about how to construct a query.

Do not specify this option from the command line when you specify the **-F parm_file** option.

If you specify the **-i** flag to query OnDemand, you cannot specify the **-q** flag.

To construct a query with a database field of type date, you must use the OnDemand internal format of the date. That is, the number of days since January 1, 1970. You can use the *arsdate* command to display the internal format for a given date. For example, to obtain the internal date for July 21, 1995, enter the following command at the AIX prompt:

```
/usr/lpp/ars/bin/arsdate -a 7/21/95
```

The *arsdate* command displays:

```
7/21/95 -> 9333
```

Enter 9333 as the index value for the date database field.

-L max_hits

Determines the maximum number of items included in the hit list, regardless of the number of items that match the query. OnDemand lists the items in the order that they were loaded in the database. Do not specify this option from the command line when you specify the **-F parm_file** option.

- n** Specifies that the items in the output file contain line numbers. If you specify this option, OnDemand sequentially numbers each line in the output file, beginning with **1**. Do not specify this option from the command line when you specify the **-F parm_file** option.

-N "(ag_name_1)...(ag_name_n)"

Determines the database fields that the `arsquery` command writes to the output file and the folder field names that appear in the header record. By default, all of the fields are included in the records in the output file. You can specify the names of one or more database fields. Each field name that you specify must be delimited with parenthesis. The entire string of field names must be enclosed in double quotes. Do not specify this option from the command line when you specify the **-F parm_file** option.

-o output_file

The name of the file where `arsquery` writes the list of items that match the query. Do not specify this option on the command line when you specify the **-F parm_file** option.

The string that you specify can contain one or more characters, except the following:

\	Backward slash	"	Double quote
'	Single quote	*	Asterisk
;	Semicolon	?	Question mark
[Left Bracket]	Right Bracket
!	Exclamation point	\$	US dollar
{	Left brace	}	Right brace
<	Less than	>	Greater than
#	Number sign	@	At sign
&	Ampersand		Vertical bar
	Blank character	-	Dash
~	Tilde		

The file name cannot exceed 255 characters.

—q "named_query"

The name of a public named query for the folder named with the **-f** flag. A named query is a set of search criteria previously saved on the library server that can be recalled by name to search a folder. A named query is typically defined to search a folder for a specific set of documents.

If you specify the **-q** flag to query OnDemand, you cannot specify the **-i** flag.

-u userid

The name of an OnDemand user. The `arsquery` command verifies that the user name that you specify is a valid OnDemand user for the server that you specify with the **-h host** option. The `arsquery` command verifies that the user name that you specify is authorized to open the folder that you specify with the **-f folder** option. If you do not specify this parameter and value, the `arsquery` command prompts you for the user name at run time.

-p password

The password of the OnDemand user that you specify with the **-u userid** option. If there is no password assigned to the user that you specify, use double quotes to show a null password. That is, specify **-p ""**. If you do not specify this parameter and value, the `arsquery` command prompts you for the password at

run time. If there is no password assigned to the user that you specify, press the Enter key when prompted.

Examples

1. The following example shows how to invoke the `arsquery` command using the command line parameters to construct a list of items and save the list in a file in the current directory. The `arsquery` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsquery -h rockies -f "student information"
-o query1.out -H
-i "WHERE student='0012-0034-0056' AND
type='B' OR type='G' OR type='T'"
```

2. The following example shows how to invoke the `arsquery` command and specify the name of a parameter file in the current directory that contains a set of queries. The result of each query is saved in a separate file in the current directory. The `arsquery` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsdocget -h rockies -F parmfile
```

The parameter file contains three queries (note the use of the AIX line continuation character):

```
[-f "student information"] [-i "WHERE type='B'"] [-o "query2.out"] \  
[-H] [-N "(student)(id)(p_date)"] \  
[-f "student information"] [-i "WHERE type='G'"] [-o "query3.out"] \  
[-H] [-N "(student)(id)(p_date)"] \  
[-f "student information"] [-i "WHERE type='T'"] [-o "query4.out"] \  
[-H] [-N "(student)(id)(p_date)"]
```

3. The following example shows how to invoke the `arsquery` command using the command line parameters to construct a hit list and send a copy of the list to a printer using the `enq` command. The `arsquery` command prompts for the OnDemand user and password.

```
/usr/lpp/ars/bin/arsquery -h rockies -f "student information"
-i "WHERE student='0012-0034-0056' AND type='T'
| enq -P G4028
```

Files

`/usr/lpp/ars/bin/arsquery`

The executable program.

Chapter 25. arstblsp command reference

Purpose

Migrate tables of application group data from the USERSPACE1 table space to their own table spaces.

Syntax

```
▶ arstblsp -a action [-d dir] -g applgrp [-p password]
▶ [-t table] [-u userid]
```

Description

The arstblsp command can be used to migrate existing tables of application group data from the USERSPACE1 table space to their own table spaces. We recommend that all existing customers (OnDemand version 2.1.1.1 or earlier) use this command to migrate application group data to table spaces, as soon as possible.

Before you can migrate application group data to table spaces, all of the tables in the application group must be closed. Use the arstblsp command with the `—1` flag to close a table.

If you plan to use ADSM to manage the log files and backup images, you must define the ADSM storage hierarchy before you migrate application group data to table spaces. Refer to the *Installation and Configuration Guide* for details.

Before you begin to migrate application group data to table spaces, we recommend that you create a full backup image of the database.

When you migrate application group data to table spaces, we strongly recommend that you do so during off-peak hours, when little or no other OnDemand activity takes place.

Parameters

—a *action* The action to perform. The action can be one of the following values:

- 0** Migrate one or more tables of application group data to their own table spaces. Specify the application group to migrate with the `—g` flag. Optionally specify a table to migrate with the `—t` flag. A table must be closed before it can be migrated to a table space.

- 1 Close a table that is still open for loading. This action causes OnDemand to close the table that is currently open for loading in the specified application group. The next time data is loaded into the application group, the data is loaded into a table space. Optionally specify a table to close with the `—t` flag.
- 2 List tables of application group data that have not been migrated to their own table spaces.
- 3 List the tables of application group data that are open for loading. An open table must be closed before it can be migrated to a table space.

—d dir Forces the `arstblsp` command to export the existing application group data to disk. If you specify this parameter, you must identify the name of a directory or filesystem with sufficient space to hold a copy of the existing application group table. Depending on the size of the table you are migrating, this can be a significant amount of disk space. For example, a table that contains 2.5 million rows, where each row is 100 bytes, may require up to 500 MB of free disk space to complete the migration.

Note: By default, OnDemand uses named pipes to export the existing application data, which provides the best performance with no filesystem overhead. We strongly encourage you to use named pipes.

—g applgrp The name of the application group to process.

—p password The password for the `userid` specified with the `—u` flag. If you do not specify this flag, the `arstblsp` command will prompt you to enter the `userid` at run time. If you specify this flag, you must also specify the `—u` flag.

—t table With the `—a1` option, identifies the table to close. With the `—a0` option, optionally identifies the table of application group data to migrate. If you do not specify this flag with the `—a0` option, the `arstblsp` command migrates all tables for the specified application group to their own table spaces.

—u userid The `userid` of a user with administrator authority for the application group. If you do not specify this flag, the `arstblsp` command will prompt you to enter the `userid` at run time. If you specify this flag, you must also specify the `—p` flag.

Examples

1. The following example shows how to list the tables of application group data that have not been migrated. In the example, the `arstblsp` command prompts for the `userid` and password of a user with administrator authority for the application group.

```
/usr/lpp/ars/bin/arstblsp -a 2 -g credit
```

2. The following example shows how to list the tables of application group data that are open for loading. In the example, the arstblsp command prompts for the userid and password of a user with administrator authority for the application group.

```
/usr/lpp/ars/bin/arstblsp -a 3 -g credit
```

3. The following example shows how to close any open tables of application group data. In the example, the arstblsp command prompts for the userid and password of a user with administrator authority for the application group.

```
/usr/lpp/ars/bin/arstblsp -a 1 -g credit
```

4. The following example shows how to close the specified table of application group data. In the example, the arstblsp command prompts for the userid and password of a user with administrator authority for the application group.

```
/usr/lpp/ars/bin/arstblsp -a 1 -g credit -t AAA26
```

5. The following example shows how to migrate tables of application group data. The arstblsp command will migrate all of the tables in the application group to their own table spaces. In the example, the arstblsp command prompts for the userid and password of a user with administrator authority for the application group.

```
/usr/lpp/ars/bin/arstblsp -a 0 -g credit
```

6. The following example shows how to migrate the specified table of application group data. In the example, the arstblsp command prompts for the userid and password of a user with administrator authority for the application group.

```
/usr/lpp/ars/bin/arstblsp -a 0 -g credit -t AAA26
```

Files

/usr/lpp/ars/bin/arstblsp

The executable program.

Notes

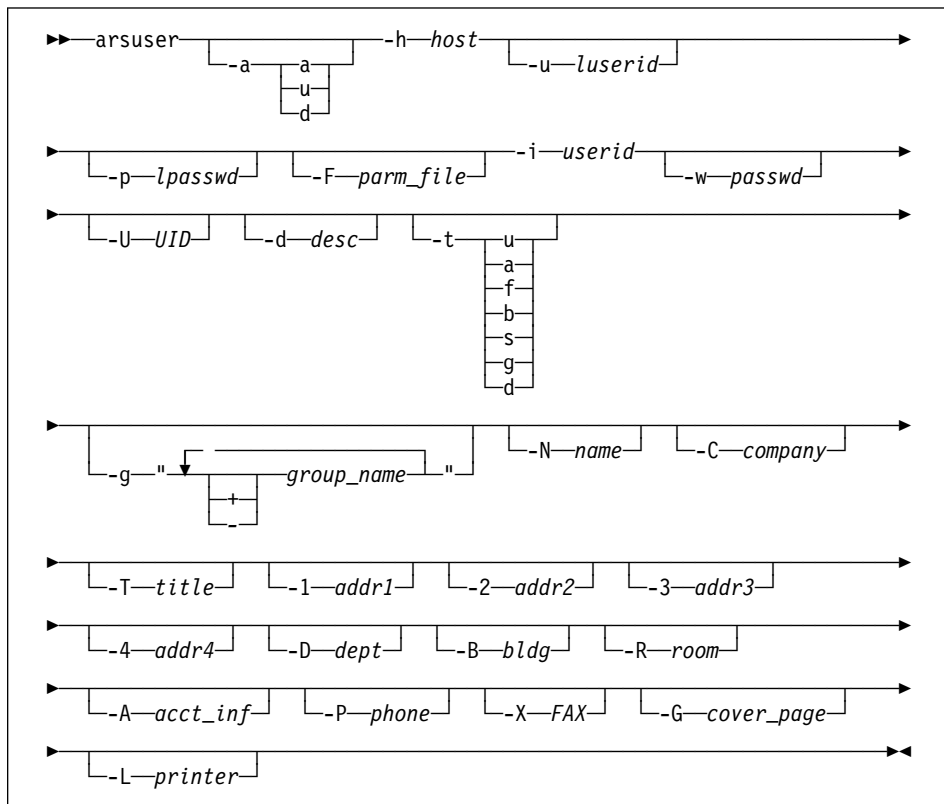
1. Before running the arstblsp command for the first time, you must relink the server programs. Refer to the arslink command in the *Installation and Configuration Guide* for details.
2. Do not load report files into OnDemand while migrating tables with the arstblsp command.

Chapter 26. arsuser command reference

Purpose

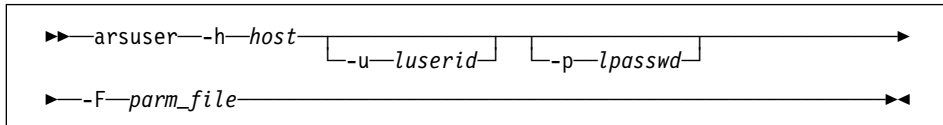
Add, update, and delete OnDemand users.

Syntax



You can specify the flags and values for the user that you want to add, update, or delete on the command line or you can specify them in a parameter file. When you use the parameter file option, specify the -h, -u, -p, and -F flags on the command line. If you do not specify the -u and -p flags on the command line, OnDemand prompts you to enter a userid and password that is valid for the server specified with the -h flag. All other flags and values required for the type of action that you want to perform must be specified in the parameter file. If you do not specify the -a flag, the default action is add a user.

The following shows the syntax using the parameter file:



Description

You can invoke the `arsuser` command from the AIX prompt or from a script file to define a user to OnDemand or update or delete an existing OnDemand user.

You specify the type of action that you want to perform, the name of the OnDemand library server where you want to update the database, and various attributes and values for the user that you are adding, updating, or deleting.

If you do not specify the type of action, the default action is to add a user. If you do not specify an OnDemand user and password for the server that you want to log on, the `arsuser` command prompts you to enter the values at run time.

When you add a user:

- If you do not specify the UID, OnDemand assigns the next available UID in the database.
- If you do not specify the user type, OnDemand sets the user type to User.
- If you do not specify the name of a default printer, OnDemand sets the default printer to *NONE.
- If you do not specify a value for a user information field, OnDemand sets the field to blank.

You can specify the name of a parameter file that contains the flags and values for one or more users that you want to add, update, or delete. Each line in the file represents a different user that you want to add, update, or delete.

Parameters

-a action

The type of action that you want the `arsuser` command to perform. Specify **a** to add a user. Specify **u** to update a user. Specify **d** to delete a user.

If you do not specify the `-a` flag, the default action is to add a user.

-h host

The host name or IP address of the OnDemand library server. The library server maintains a database of information about OnDemand users.

This is a required parameter and can only be specified on the command line.

-u userid

The name of an OnDemand user to log on the server. The `arsuser` command verifies that the user name that you specify is valid for the server that you specify

with the **-h host** flag. The `arsuser` command verifies that the user name that you specify is authorized to add users to OnDemand.

If you do not specify this flag and value, the `arsuser` command prompts you for the user name at run time.

-p lpasswd

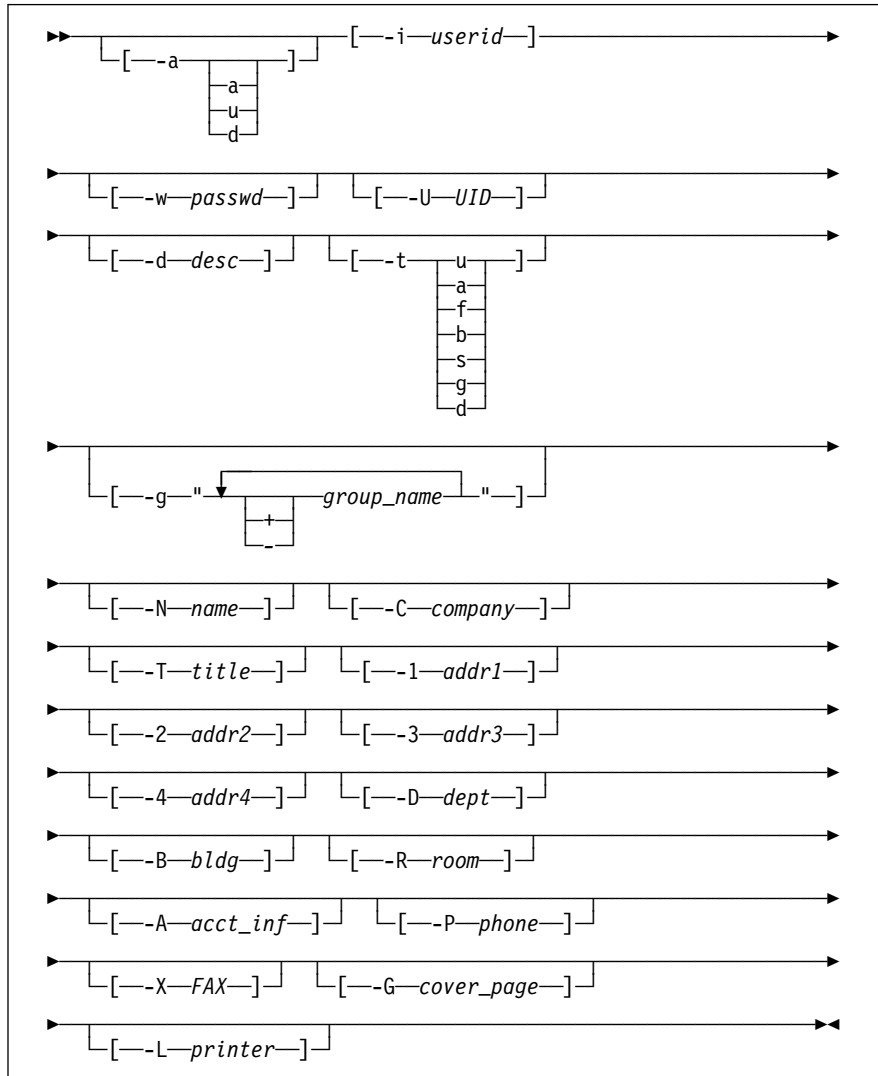
The password of the OnDemand user that you specify with the **-u luserid** flag. If there is no password assigned to the user that you specify, use double quotes to show a null password. That is, specify **-p ""**.

If you do not specify this flag and value, the `arsuser` command prompts you for the password at run time. If there is no password assigned to the user that you specify, press the Enter key when prompted.

-F parm_file

The name of a file that contains the flags and values for one or more users that you are adding, updating, or deleting.

The following shows the flags and values that you can specify in the parameter file:



The flags and values in the file must be enclosed in brackets. You must specify the `-i` flag. Each line in the file represents a different user that you want to add, update, or delete.

-i userid

The `userid` that you want to add, update, or delete. The `userid` can be from 1 to 20 characters in length. When you add a user, OnDemand verifies that the `userid` is not already defined in the database. Before updating or deleting a user, OnDemand verifies that the `userid` is defined in the database.

This is a required parameter that can be specified on the command line or in the parameter file.

-w password

The password for the userid that you want to add or update. The password can be from 1 to 20 characters in length. When you add a user, you must specify this flag and value. When you update a user, this flag is optional.

If you do not want to assign a password to the user, you must use double quotes to show a null password. That is, specify **-w ""**.

-U UID

The UID to assign the user, when adding a user. If you do not specify this flag and value, OnDemand assigns the next available UID in the database. If you assign a UID, OnDemand verifies that the value that you specify is valid and is not already defined in the database.

-d desc

You can enter up to 120 characters of descriptive information about the user. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-d "This is a description"**.

-t type

This is the user type. If you do not specify this flag when you add or update a user, the default user type is user (u). OnDemand supports the following user types:

- u** User
- a** Create application groups and applications
- f** Create folders
- b** Create application groups, applications, and folders
- s** System administrator
- g** User administrator
- d** Application group and folder administrator

-g group_specification

Assign the user to or remove the user from one or more groups with this flag. You can remove a user from a group only during the update action.

Specify **-g group** or **-g +group** to assign the user to a group. Specify **-g -group** to remove the user from a group. If you do not specify a + (plus) or a - (minus) before the group name, the default action is to add the user to the group.

You can add the user to one or more groups in a single action. For example, the following fragment:

```
-g "security +admin"
```

Causes OnDemand to add the user to the security and admin groups.

If you are updating a user, you can add the user to or remove the user from one or more groups in a single action. For example, the following fragment:

```
-g "mgmt -security -admin"
```

Causes OnDemand to add the user to the mgmt group and remove the user from the security and admin groups.

-N name

The user's name. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-N "J. Kelvin, X 5555"**.

-C company

The name of the company. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-C "Acme Unlimited"**.

-T title

The user's title or job responsibility. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-T "Special Operations"**.

-1 addr1

The first of four address lines. You can enter 60 characters of information in an address line. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-1 "Jessica Kelvin"**.

The OnDemand CICS/ESA client program places the information that you enter in the first address line on the print separator page.

-2 addr2

The second of four address lines. You can enter 60 characters of information in an address line. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-2 "Mail Drop G16"**.

-3 addr3

The third of four address lines. You can enter 60 characters of information in an address line. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-3 "6124 Ruatan Place"**.

-4 addr4

The fourth of four address lines. You can enter 60 characters of information in an address line. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-4 "Iowa City IA 77777-0001"**.

-D dept

The user's department name or number. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special

characters, you must enclose the string in quotes. For example, **-D "G16 Purchasing"**.

-B bldg

The user's building or location. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-B "Rusty Bucket"**.

-R room

The user's room number. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-R "001 4"**.

-A acct_inf

Accounting or other related information. You can enter from 1 to 60 characters of information. If the information that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-A "G16-0110 P95 A"**.

OnDemand includes the string that you specify in system log messages generated by the actions performed by the user. These messages are also passed to the system log user exit shell script.

-P phone

The user's phone number. The phone number can be from 1 to 32 characters in length. If the phone number that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-P "303 555 1212"**.

-X FAX

The phone number of the user's FAX machine. The phone number can be from 1 to 32 characters in length. If the phone number that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-X "303 555 1212"**.

-G cover_page

The name of an AFP file that contains a user-defined FAX cover page. The file name can be from 1 to 8 characters in length.

-L printer

The name of the default server printer for the user. The printer name can be from 1 to 60 characters in length. If the name that you enter contains the blank or other special characters, you must enclose the string in quotes. For example, **-L "Dallas 3990"**.

Examples

1. The following example shows how to invoke the `arsuser` command from the AIX prompt to add a user to the database.

```
/usr/lpp/ars/bin/arsuser -h rockies -u admin -p ""  
-i csr0001 -w "" -N "Earl Roy"
```

2. The following example shows how to invoke the `arsuser` command and specify the name of a parameter file to add a user to the database.

```
/usr/lpp/ars/bin/arsuser -h rockies -u admin -p ""  
-F parmfile
```

The parameter file contains the information that OnDemand uses to add the user.

```
[-i csr0001] [-w ""] [-N "Earl Roy"]
```

3. The following example shows how to invoke the `arsuser` command from the AIX prompt to add a user to the database. The `arsuser` command prompts for the userid and password to log on to the server.

```
/usr/lpp/ars/bin/arsuser -h rockies -i csr0001  
-w "" -N "Earl Roy"
```

4. The following example shows how to invoke the `arsuser` command from the AIX prompt to add an administrator user to the database and assign a default printer for the user. The `arsuser` command prompts for the userid and password to log on to the server.

```
/usr/lpp/ars/bin/arsuser -h rockies -i admin01 -w ""  
-t s -N "Earl Roy" -L P4028
```

5. The following example shows how to invoke the `arsuser` command and specify the name of a parameter file to add several users to the database.

```
/usr/lpp/ars/bin/arsuser -h rockies -u admin -p ""  
-F parmfile
```

The parameter file contains the information that OnDemand uses to add the users, including the name of the OnDemand group to assign the users.

```
[-i csr0001] [-w ""] [-N "Earl Royce"] [-g "+csr"]
[-i csr0002] [-w ""] [-N "Pearl Aman"] [-g "+csr"]
[-i csr0003] [-w ""] [-N "June James"] [-g "+csr"]
[-i csr0004] [-w ""] [-N "Grace Kelp"] [-g "+csr"]
[-i csr0005] [-w ""] [-N "James June"] [-g "+csr"]
```

6. The following example shows how to invoke the `arsuser` command from the AIX prompt to update a user, changing the user type to user administrator. The `arsuser` command prompts for the `userid` and `password` to log on to the server.

```
/usr/lpp/ars/bin/arsuser -a u -h rockies -i csr0001 -t g
```

7. The following example shows how to invoke the `arsuser` command from the AIX prompt to delete a user.

```
/usr/lpp/ars/bin/arsuser -h rockies -u admin -p ""
-a d -i admin2
```

Files

`/usr/lpp/ars/bin/arsuser`

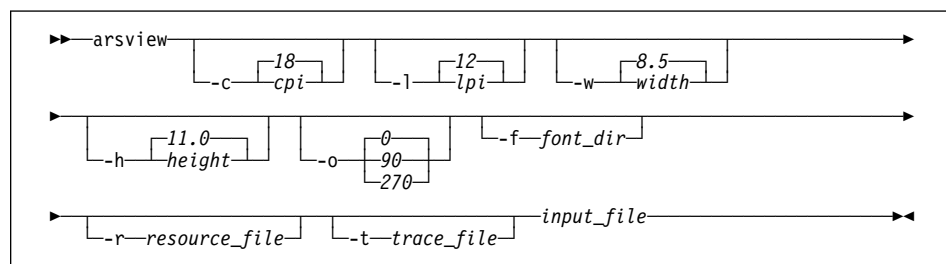
The executable program.

Chapter 27. arsview command reference

Purpose

The arsview command reads an AFP document and creates a line data file suitable for viewing on a 3270 or other line-mode display.

Syntax



Description

The OnDemand CICS/ESA client program displays items retrieved from the database on 3270, line-mode terminals. If the selected item contains AFP data, OnDemand first calls the arsview command to read the AFP data, process the command options, and create a line data file. OnDemand then passes the line data file to the client program. The arsview command can also be used as a general purpose AFP to line data batch conversion utility. You can invoke the arsview command from within a script file or type the command at the AIX prompt.

You can optionally specify the directory where the arsview command can locate user-defined character set and code page files to process the input file. Refer to the *Installation and Configuration Guide* for information about defining character set and code page files for use with OnDemand client programs.

The arsview command ignores any graphics, images, and typographic fonts in the input file. All fonts in the input file are mapped to a single point, monospaced font. You can specify the name of the resource group file that contains the form definition and overlay associated with the input file. The arsview command can process information contained in the form definition, such as the offset, and write the overlay text to the output file.

Parameters

-c cpi Characters per inch in the horizontal direction of the smallest (or only) font in the input file. For example, if the input file contains text at 12 characters per inch and 10 characters per inch, you would specify **-c 12**. The default value is 18 characters per inch.

-l lpi	Lines per inch in the vertical direction of the smallest (or only) font in the input file. The default value is 12 lines per inch.
-w width	The width of a page in the input file. The -w and -h values define the paper size used to print a page of the input file. The default value is 8.5 inches.
-h height	The height of a page in the input file. The -w and -h values define the paper size used to print a page of the input file. The default value is 11.0 inches.
-o orientation	The orientation of the data on the page in the input file. You can specify that the data appears rotated 0, 90, or 270 degrees. The default value is 0 degrees.
-f font_dir	The name of the directory that contains user-defined character set and code page files required to process the input file. By default, the <i>arsview</i> command searches the <code>/usr/lpp/ars/font</code> directory. Refer to the <i>Installation and Configuration Guide</i> for information about defining character set and code page files for use with OnDemand client programs.
-r resource_file	The name of the resource group file that contains the form definition and overlay associated with the input file. The <i>arsview</i> command can process information contained in the form definition, such as the offset, and include text from an overlay in the output file.
-t trace_file	Specifies the name of the file where the <i>arsview</i> command writes debugging information. You can use this option to obtain detailed information about how the <i>arsview</i> command translates the input file.
input_file	The name of the source (AFP) file.

Examples

1. The following example shows how to invoke the *arsview* command from the AIX prompt. The *arsview* command reads the file `37271600.AFP` (located in the current directory) and the options specified and redirects the line data output to the file `37271600.LIN` in the current directory. By default, the input file contains 8.5 x 11 inch pages.

```
/usr/lpp/ars/bin/arsview -c 12 -l 8 -o 90
37271600.AFP > 37271600.LIN
```

2. In the following example, the *arsview* command processes an input file using the default options for characters per inch, lines per inch, width, height, and orientation. The output is redirected to stdout.

```
/usr/lpp/ars/bin/arsview 96089900.AFP
```

Files

`/usr/lpp/ars/bin/arsview` The executable program.

Index

A

- about
 - this publication xvii
- accessing
 - files 20
 - image files 20
 - index data 5
 - reports 5
 - resources 5
- adding space to the ADSM database 81
- adding space to the ADSM recovery log 81
- adding storage volumes
 - optical library 71
 - tape backup 79
 - to a library 71
- ADSM for AIX
 - adding optical storage volumes 71
 - adding space to the database 81
 - adding space to the recovery log 81
 - adding storage volumes 71
 - adding tape storage volumes 79
 - capacity of database 83
 - capacity of recovery log 83
 - check in library volume command 73
 - check out library volume command 78
 - database
 - adding space to 81
 - backup and recovery 65
 - capacity 83, 84
 - defining disk volumes 82
 - defining logical volumes 81
 - determine capacity 83
 - disk volumes 82
 - displaying capacity 84
 - extending capacity 84
 - logical volumes 81
 - database recovery 67
 - defining disk volumes 82
 - defining logical volumes 81
 - determine capacity of database 83
 - determine capacity of recovery log 83
 - device configuration file 67
 - displaying capacity of database 84
 - displaying capacity of recovery log 84
 - dsmlabel command 71, 79
 - ADSM for AIX (*continued*)
 - extending capacity of database 84
 - extending capacity of recovery log 84
 - halting 39
 - logical volumes 81
 - query volume command 76
 - query volume command detail 76
 - recovering the database 67
 - recovery log 66
 - adding space to 81
 - capacity 83, 84
 - determine capacity 83
 - displaying capacity 84
 - extending capacity 84
 - removing optical storage volumes 75
 - removing storage volumes 75
 - report file backup and recovery 68
 - restarting 34
 - starting 33, 34
 - stopping 36, 39
 - volume history file 67
- ADSM for AIX Query Database Command 50
- ADSM for AIX Query Volume Command 50
- AIX Performance Monitoring and Tuning Guide 55
- AIX Performance Toolbox/6000 54
- AIX/6000 performance and tuning 53
- alerts 41
- allocating storage space
 - for data loading 4
- API 43
 - See also* application programming interface
- application
 - DATASET parameter 112, 147
 - requirements for data loading 4, 18
- application group
 - FORMS parameter 112, 147
 - requirements for data loading 4, 17
- application programming interface
 - system logging user exit shell script 43
 - user exit shell script 43
- applications
 - loading image files 17
 - loading index data 3, 17
 - loading reports 3
 - storing data 3, 17
- ars_adsm

- ars_adsm (*continued*)
 - ADSM database backup 14
 - backup the ADSM database 14
 - command reference 99
 - database backup 14
- ars_db
 - backup the OnDemand database 13
 - command reference 103
 - database backup 13
 - OnDemand database backup 13
- ars_load
 - checking messages 15
 - command reference 111
 - example 22
 - messages 15
 - setting up 6, 21
 - starting 33
- ars_load Command initialization Record 6
- arsadmin
 - command reference 117
 - unloading a report 27
- arsdate
 - command reference 129
- arsdocget
 - command reference 133
- arsjesd
 - command reference 145
- arslog
 - system log user exit shell script 43
 - user exit 43
- arslog.create
 - initializing the system log 42
- arsmaint
 - command reference 149
- arsquery
 - command reference 155
- arstblsp
 - command reference 161
- arsuser
 - command reference 165
- arsview
 - command reference 175

B

- backing up databases
 - after loading reports 13
- backup and recovery
 - ADSM database 67
 - ADSM device configuration file 67

- backup and recovery (*continued*)
 - ADSM for AIX database 65
 - ADSM recovery log 66
 - ADSM volume history file 67
 - control files 59
 - database 60, 63
 - DB2 for AIX 60
 - OnDemand 59
 - OnDemand database 60, 63
 - overview 59
 - product software 59
 - recovery log 66
 - report files and resources 68
- backup the ADSM database 14
- backup the OnDemand database 13
- base operating system performance package 53
- batch processing
 - arsdocget 133
 - arsquery 155
 - create a document list 155
 - create a hit list 155
 - retrieve documents from the database 133
 - retrieve items from the database 133

C

- cache file systems
 - expiration processing 149
 - migration processing 149
 - monitoring utilization 48
- CD-ROM creation manifest
 - reviewing 43
- changing files xx
- Check In Library Volume Command 73
- check messages and verify totals
 - ars_load processing 15
- Check Out Library Volume Command 78
- column delimiter 125
- command reference
 - overview 93
- commands 47
 - ADSM for AIX
 - check in library volume 73
 - check out library volume 78
 - dsmlabel 71, 79
 - query volume 76
 - query volume detail 76
 - ars_adsm 99
 - ars_db 103
 - ars_load 111

commands (*continued*)

- arsadmin 117
- arsdate 129
- arsdocget 133
- arsjesd 145
- arslog 43
- arsmaint 149
- arsquery 155
- arstblsp 161
- arsuser 165
- arsview 175
- compress
 - arsadmin function 119
- cron 62
- decompress
 - arsadmin function 119
- df 48
- get database configuration 55
- get database manager 55
- get snapshot 55
- how to read a syntax diagram 95
- import
 - arsadmin function 118
- indexer
 - arsadmin function 120
- iostat 49
- load
 - arsadmin function 117
- load_db
 - arsadmin function 117
- lsps 48
- netstat 51
- postprocess_cmd
 - arsadmin function 120
- ps 45, 46
- query volume 50
- res_diff
 - arsadmin function 119
- retrieve
 - arsadmin function 118
- starting ADSM for AIX 34
- starting ars_load 35
- starting arsjesd 34
- starting arsobjd 35
- starting arsockd 35
- starting DB2 for AIX 34
- starting the data loading daemon 35
- starting the server daemon 35
- stopping ADSM for AIX 38
- stopping DB2 for AIX 37

commands (*continued*)

- store
 - arsadmin function 118
- tar 60
- unload
 - arsadmin function 117
- uptime 47
- vmstat 47
- compress
 - arsadmin command function 119
- compression
 - estimating 119
- configuration
 - MVS Download feature xvii
- conventions
 - AIX xx
 - changing files xx
 - editing files xx
 - highlighting xx
 - text editor xx

D

- data
 - download user exits 29
 - indexing on the MVS system 8
 - indexing on the RS/6000 7, 20
 - loading 3, 17
 - loading user exits 29
 - MVS download feature 5
 - MVS download processing 29
- database
 - ADSM
 - adding space to 81
 - backup 13, 61
 - backup image 62
 - capacity 83, 84
 - determine capacity 83, 84
 - expiration processing 149
 - expiring index data 87
 - extending capacity 84
 - importing index data 87, 118
 - importing tables 118
 - index maintenance 87
 - loading index data 11
 - logging 62
 - maintaining index data 87
 - migrating index data 87
 - migration 87, 118
 - migration processing 149

- database (*continued*)
 - recovery 63
 - segment table 11
- DATASET parameter 112
 - application 147
 - ars_load command 147
- date
 - internal format, obtaining 129
 - obtaining OnDemand internal format 129
 - OnDemand internal format, obtaining 129
- DB2 for AIX
 - backup and recovery 60
 - configuration parameters 56
 - database system monitor 55
 - get database configuration command 55
 - get database manager command 55
 - get snapshot command 55
 - halting 39
 - index statistics 56
 - performance and tuning 55
 - processes 45
 - restarting 34
 - starting 33, 34
 - stopping 36, 39
- decompress
 - arsadmin command function 119
- defining disk volumes 82
- defining logical volumes 81
- deleting a report 27
- deleting index data 87
- determine capacity of database 83
- determine capacity of recovery log 83
- df Command 48
- df Command Output 49
- disk volumes
 - defining for ADSM 82
- displaying capacity of database 84
- displaying capacity of recovery log 84
- document list
 - batch processing 155
- documentation related to OnDemand xx
- dsmdb logical volume 81
- dsmlabel command 71, 79
- dsmlog logical volume 81

E

- editing files xx
- errors and alerts 41

- estimating and allocating storage space
 - for data loading 4
- expiration processing 149
- expiring index data 87
- extending capacity of database 84
- extending capacity of recovery log 84

F

- file systems
 - monitoring 48
- folder
 - requirements for data loading 19
 - system log 42
- FORMS parameter
 - application group 147
 - ars_load command 147
 - in ars_load command 112
- Full Backup of the ADSM Database 14

G

- generic index data
 - creating 20
- generic index file 21
 - creating 20
- Get Database Configuration Command 55
- Get Database Manager Command 55
- Get Snapshot Command 55

H

- halting the server 36
- highlighting conventions xx
- hit list
 - batch processing 155
- how to read a syntax diagram 95

I

- IBM trademarks used in this book xv
- image data
 - indexing on the RS/6000 20
- image file
 - indexing on the RS/6000 20
 - loading 17
 - processing 22
 - storing 17
- import
 - arsadmin command function 118

- import (*continued*)
 - index data from archive media 118
- importing index data 87
- index data
 - accessing 5
 - creating 20
 - downloading 5
 - loading 3, 17
 - loading in the database 11
 - loading the database 3, 17
 - processing 10, 22
 - transferring 5
- index file
 - creating 20
- index maintenance
 - database 87
 - expiring index data 87
 - importing 87
 - maintaining index data 87
 - migration 87
- indexer
 - arsadmin command function 120
- indexing
 - restarting a load process 25
- indexing files on the RS/6000 20
- Indexing Reports on MVS 9
- indexing reports on the MVS system 8
- indexing reports on the RS/6000 7, 8
- inittab
 - ars_load command 6
- inittab Records for OnDemand Programs 33
- installation
 - MVS Download feature xvii
- internal date format, obtaining 129
- iostat Command 49
- iostat Command Output 50
- iostat Continuous Command 50

L

- Labeling Storage Volumes 71
- Labeling Tape Storage Volumes 79
- legal notices xv
- load
 - arsadmin command function 117
- load_db
 - arsadmin command function 117
- loading data 1, 3, 17
 - accessing
 - files 20
 - image files 20

- loading data (*continued*)
 - accessing (*continued*)
 - index data 5
 - reports 5
 - resources 5
 - application group requirements 4, 17
 - application requirements 18
 - ars_load command 6, 21
 - folder requirements 19
- loading image files 17
- loading index data 3, 17
- loading index rows in the database 11
- loading reports 1, 3
 - preparing 4
- loading storage objects on storage volumes 12
- logging facility 42
 - API definition 43
 - user exit shell script 43
- logical volumes
 - allocating space for 81
 - defining for AD SM 81
 - dsmdb 81
 - dsmlog 81
- lsps Command 48
- lsps Command Output 48

M

- maintaining index data 87
- manifest
 - reviewing CD-ROM contents 43
- messages
 - ars_load processing 15, 23
 - system logging facility 15, 23
 - viewing the system log 42
- migrating
 - table spaces 161
- migrating index data 87
- migration 118
- migration processing 149
- monitoring
 - AD SM for AIX processes 46
 - cache file systems 48
 - DB2 for AIX processes 45
 - disks and file systems 48
 - file systems 48
 - I/O activity 49
 - memory usage 47
 - network 51
 - OnDemand processes 44

- monitoring (*continued*)
 - processes 44
 - server 41
 - storage volumes 50
 - system load 47
 - system paging space 48
- Monitoring ADSM for AIX Processes with the ps Command 47
- Monitoring DB2 for AIX Processes with the ps Command 46
- Monitoring OnDemand Processes with the ps Command 45
- MVS download feature
 - about xvii
 - accessing data 5
 - arsjesd command reference 145
 - exits 29
 - processing 29
 - restarting 34
 - starting 33, 34
 - starting related programs 35
 - stopping 36
 - transmitting data 5

N

- netstat Command 51
- netstat Command Output 51
- notices xv

O

- obtaining OnDemand internal date format 129
- Offline Backup of the OnDemand Database 14
- Offline Database Backup Command 62
- OnDemand
 - alerts 41
 - command reference 93
 - documentation xx
 - errors and alerts 41
 - halting 38
 - logging facility 42
 - operating the server 31
 - product support xviii
 - starting the data loading daemon 35
 - starting the server daemon 35
 - stopping the server daemon 38
 - system logging facility 42
- OnDemand database
 - backup image 62

- OnDemand database (*continued*)
 - deleting database indexes 87
 - expiring database indexes 87
 - index maintenance 87
 - maintaining index data 87
 - migrating database indexes 87
 - recovery 63
- OnDemand internal date format, obtaining 129
- operating the server
 - adding storage volumes 71
 - adding storage volumes for tape backup 79
 - backup and recovery 59
 - monitoring 41
 - optical storage volumes, adding 71
 - optical storage volumes, removing 75
 - overview 31
 - performance and tuning 53
 - recovery 59
 - removing storage volumes 75
 - restarting AIX 39
 - restarting OnDemand 39
 - restarting the RS/6000 39
 - restarting the server 39
 - starting 33
 - starting related MVS programs 35
 - storage volumes, adding 71
 - storage volumes, removing 75
 - tape backup 79
- other companies trademarks used in this book xv

P

- performance and tuning
 - base operating system 53
 - DB2 for AIX 55
 - OnDemand server 56
 - overview 53
 - TCP/IP 56
- postprocess_cmd
 - arsadmin command function 120
- preparing to load reports 4
- processes
 - ADSM for AIX 46
 - monitoring 44
 - OnDemand 44
- processing image files 22
- processing index data and image files 22
- Processing Index Data and Reports 10
- processing index data, reports, and resources 10

product support xviii

Q

Query Database Command 50
Query Volume Command 50, 76
Query Volume Command Detail 76
Query Volume Command Detail Output 77
Query Volume Command Output 76

R

rebooting the RS/6000 36
recovering the ADSM database 67
recovery 59
 See also backup and recovery
recovery log
 ADSM
 adding space to 81
 capacity 83, 84
 determine capacity 83, 84
 extending capacity 84
reinitializing AIX/6000 36
related documentation xx
related IBM products
 MVS download feature xvii
removing a report 27
removing storage volumes
 from a library 75
 optical library 75
report
 accessing 5
 deleting from OnDemand 27
 indexing on the MVS system 8
 loading 3
 processing 10
 removing from OnDemand 27
 restarting a load process 25
 storing 3
 unloading from OnDemand 27
report file backup and recovery 68
reports
 indexing on the RS/6000 7
 preparing to load 4
res_diff
 arsadmin command function 119
resource file
 input to arsadmin load command 125
 input to arsadmin res_diff command 126
 processing 10

resources
 accessing 5
 downloading 5
 transferring 5
restarting a load process 25
restarting ADSM 34
restarting DB2 for AIX 34
restarting OnDemand 39
restarting related MVS programs 35
restarting the server 39
retrieve
 arsadmin command function 118
reviewing a CD-ROM creation manifest 43

S

searching for and viewing messages 42
segment table 11
server
 backup and recovery 59
 monitoring 41
 performance 53
 recovery 59
 starting 33
 tuning 53
server daemon
 starting 35
 stopping 38
service marks used in this book
 IBM xv
 other companies xv
setting up the ars_load command 6, 21
shutting down AIX/6000 36
shutting down OnDemand 36
starting ADSM 34
starting ars_load 35
starting arsjesd 34
starting arsockd 35
starting darsobjd 35
starting DB2 for AIX 34
starting MVS download 34
starting OnDemand 33
starting related MVS programs 35
starting the data loading daemon 35
starting the MVS download feature 33
starting the server daemon 35
stopping ADSM 39
Stopping ADSM for AIX 38
Stopping DB2 for AIX 37, 39

- stopping OnDemand 36
- stopping the MVS download feature 36
- stopping the OnDemand server daemon 38
- stopping the server 36
- storage objects
 - loading on storage volumes 12
- store
 - arsadmin command function 118
- syntax diagram
 - how to read 95
- system log folder 42
- system logging facility 42
 - API definition 43
 - ars_load messages 15, 23
 - messages 15, 23
 - user exit shell script 43

T

- table spaces
 - migrating 161
- tables
 - segment 11
- tar Command 60
- TCP/IP
 - performance and tuning 56
- text editor xx
- TIFF file
 - loading 17
 - storing 17
- trademarks used in this book
 - IBM xv
 - other companies xv

U

- unload
 - arsadmin command function 117
- unloading a report 27
- uptime Command 47
- uptime Command Output 47
- user exit shell script 43
- user exits
 - data download 29
 - data loading 29
 - downloading data 29
 - loading data 29
 - MVS download 29

V

- vmstat Command 47
- vmstat Command Output 48

Readers' Comments — We'd Like to Hear from You

**OnDemand for AIX
Administrator's Reference
Version 2.2**

Publication No. S544-5293-01

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

Note: IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

	Yes	No
• Does the publication meet your needs?	_____	_____
• Did you find the information:		
Accurate?	_____	_____
Easy to read and understand?	_____	_____
Easy to retrieve?	_____	_____
Organized for convenient use?	_____	_____
Legible?	_____	_____
Complete?	_____	_____
Well illustrated?	_____	_____
Written for your technical level?	_____	_____
• Do you use this publication:		
As an introduction to the subject?	_____	_____
As a reference manual?	_____	_____
As an instructor in class?	_____	_____
As a student in class?	_____	_____
• What is your occupation?	_____	_____

Thank you for your input and cooperation.

Note: You may either send your comments by fax to 1-800-524-1519, or mail your comments. If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

Comments:

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
S544-5293-01



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



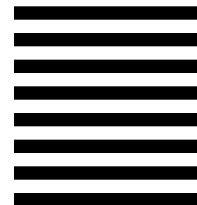
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
The IBM Printing Systems Company
Department H7FE Building 003G
P O Box 1900
BOULDER CO 80301-9817



Fold and Tape

Please do not staple

Fold and Tape

S544-5293-01

Cut or Fold
Along Line



Part Number: 4071160
Program Number: 5622-662



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

S544-5293-01



Spine information:



OnDemand for AIX

Administrator's Reference

Version 2.2