

ADSTAR Distributed Storage Manager  
for OS/2



# Administrator's Reference

*Version 2*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

**First Edition (September 1996)**

This edition applies to Version 2 Release 1 of the ADSTAR Distributed Storage Manager (84H3121), and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation  
Information Development, Dept. 61C  
9000 South Rita Road  
Tucson, AZ 85744-0001, U.S.A.

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.

© **Copyright International Business Machines Corporation 1996. 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

<b>Notices</b> . . . . .	xi
Trademarks . . . . .	xii
<b>Preface</b> . . . . .	xv
Who Should Read This Publication . . . . .	xv
What You Should Know Before Reading This Publication . . . . .	xv
ADSTAR Distributed Storage Manager Publications . . . . .	xv
Software Developer's Program . . . . .	xvi
Translations . . . . .	xvii
Do You Have Comments or Suggestions? . . . . .	xvii
<b>Changes for ADSTAR Distributed Storage Manager Version 2</b> . . . . .	xix
<b>Chapter 1. Using Syntax Diagrams</b> . . . . .	1
<b>Chapter 2. Entering Administrative Commands</b> . . . . .	5
Command Components . . . . .	5
Command Name . . . . .	5
Positional Parameters . . . . .	5
Keyword Parameters . . . . .	5
Command Entry Rules . . . . .	6
Processing Administrative Commands . . . . .	7
Commands Processed in the Foreground . . . . .	7
Commands Processed in the Background . . . . .	7
Formatting Command Output on a Terminal . . . . .	8
Using Wildcard Characters . . . . .	8
Examples for Match-Any Characters . . . . .	9
Examples for Match-Exactly-One Character . . . . .	10
Naming Conventions . . . . .	10
Naming Volumes . . . . .	10
Case-Sensitivity . . . . .	10
Naming Restrictions for Objects . . . . .	11
<b>Chapter 3. Using the Command Line Interface</b> . . . . .	13
Using the Server Console Session . . . . .	13
Using the Administrative Client Session . . . . .	13
Using Console Mode . . . . .	14
Using Mount Mode . . . . .	15
Using Batch Mode . . . . .	16
Using Interactive Mode . . . . .	16
Using Administrative Client Options . . . . .	18
Using Continuation Characters . . . . .	19
Redirecting Command Output to a File in Batch and Interactive Modes . . . . .	19
Using Macros to Issue Commands . . . . .	21

<b>Chapter 4. Using Commands Based on Privilege Class</b>	29
Commands Requiring System Privilege	29
Commands Requiring Policy Privilege	33
Commands Requiring Storage Privilege	35
Commands Requiring Operator Privilege	38
Commands Requiring Analyst Privilege	39
Commands Any Administrator Can Issue	40
<b>Chapter 5. Using Administrator Commands</b>	45
ACTIVATE POLICYSET (Activate a New Policy Set)	46
ASSIGN DEFMGMTCLASS (Assign a Default Management Class)	48
AUDIT LIBRARY (Verify an Automated Library's Inventory)	51
AUDIT LICENSES (Audit Data Storage Usage)	54
AUDIT VOLUME (Verify Database Information for a Storage Pool Volume)	56
BACKUP DB (Back Up the Database)	61
BACKUP DEVCONFIG (Create Backup Copies of Device Configuration Information)	66
BACKUP STGPOOL (Back Up Primary Storage Pool to Copy Storage Pool)	68
BACKUP VOLHISTORY (Save Sequential Volume History Information)	72
CANCEL PROCESS (Cancel an Administrative Process)	74
CANCEL REQUEST (Cancel One or More Mount Requests)	76
CANCEL SESSION (Cancel One or More Client Sessions)	78
CHECKIN LIBVOLUME (Check a Storage Volume into a Library)	80
CHECKOUT LIBVOLUME (Check a Storage Volume out of a Library)	84
COMMIT (Control Committing of Commands in a Macro)	87
COPY DOMAIN (Copy a Policy Domain)	89
COPY MGMTCLASS (Copy a Management Class)	92
COPY POLICYSET (Copy a Policy Set)	94
COPY SCHEDULE (Copy a Client or an Administrative Command Schedule)	96
Copying Client Schedules	96
Copying Administrative Command Schedules	97
DEFINE ASSOCIATION (Associate Client Nodes with a Schedule)	100
DEFINE COPYGROUP (Define a Copy Group)	103
DEFINE DBBACKUPTRIGGER (Define the Database Backup Trigger)	112
DEFINE DBCOPY (Define a Volume Copy of a Database Volume)	116
DEFINE DBVOLUME (Define a Database Volume)	118
DEFINE DEVCLASS (Define a Device Class)	120
Syntax for Defining a Device Class with a Device Type of 4MM	121
Syntax for Defining a Device Class with a Device Type of 8MM	124
Syntax for Defining a Device Class with a Device Type of DLT	127
Syntax for Defining a Device Class with a Device Type of FILE	130
Syntax for Defining a Device Class with a Device Type of LAN3995	132
Syntax for Defining a Device Class with a Device Type of MO35	133
Syntax for Defining a Device Class with a Device Type of OPTICAL	136
Syntax for Defining a Device Class with a Device Type of QIC	138
Syntax for Defining a Device Class with a Device Type of REMOVABLEFILE	141
DEFINE DOMAIN (Define a New Policy Domain)	144
DEFINE DRIVE (Define a Drive to a Library)	147
DEFINE LIBRARY (Define a Library)	150

DEFINE LOGCOPY (Define a Volume Copy of a Recovery Log Volume)	153
DEFINE LOGVOLUME (Define a Recovery Log Volume)	155
DEFINE MGMTCLASS (Define a Management Class)	157
DEFINE POLICYSET (Define a Policy Set)	161
DEFINE SCHEDULE (Define a Client or an Administrative Command Schedule)	164
Defining Client Schedules	164
Defining Administrative Command Schedules	171
DEFINE STGPOOL (Define a Storage Pool)	180
DEFINE VOLUME (Define a Volume in a Storage Pool)	193
DELETE ASSOCIATION (Delete Node Associations with a Schedule)	197
DELETE COPYGROUP (Delete a Copy Group)	199
DELETE DBBACKUPTRIGGER (Delete the Database Backup Trigger)	202
DELETE DBVOLUME (Delete a Database Volume)	204
DELETE DEVCLASS (Delete a Device Class)	206
DELETE DOMAIN (Delete a Policy Domain)	208
DELETE DRIVE (Delete a Drive from a Library)	210
DELETE EVENT (Delete Event Records)	212
DELETE FILESPACE (Delete Client Node Data from the Server)	216
DELETE LIBRARY (Delete a Library)	220
DELETE LOGVOLUME (Delete a Recovery Log Volume)	222
DELETE MGMTCLASS (Delete a Management Class)	224
DELETE POLICYSET (Delete a Policy Set)	226
DELETE SCHEDULE (Delete a Client or an Administrative Command Schedule)	229
Deleting Client Schedules	229
Deleting Administrative Command Schedules	229
DELETE STGPOOL (Delete a Storage Pool)	232
DELETE VOLHISTORY (Delete Sequential Volume History Information Collected by the Server)	234
DELETE VOLUME (Delete a Storage Pool Volume)	240
DISABLE (Temporarily Prevent Client Node Access to the Server)	243
DISMOUNT VOLUME (Dismount a Volume by Volume Name)	245
ENABLE (Resume User Activity on the Server)	246
EXPIRE INVENTORY (Manually Start Inventory Expiration Processing)	248
EXPORT ADMIN (Copy Administrator Information to Sequential Media)	250
EXPORT NODE (Copy Client Node Information to Sequential Media)	254
EXPORT POLICY (Copy Policy Information to Sequential Media)	260
EXPORT SERVER (Copy Server Information to Sequential Media)	264
EXTEND DB (Increase the Assigned Capacity of the Database)	269
EXTEND LOG (Increase the Assigned Capacity of the Recovery Log)	271
GRANT AUTHORITY (Add Administrator Authority)	273
HALT (Shut Down the Server)	277
HELP (Get Help on Commands and Error Messages)	280
IMPORT ADMIN (Import Administrator Information)	283
IMPORT NODE (Import Client Node Information)	287
IMPORT POLICY (Import Policy Information)	293
IMPORT SERVER (Import the Server)	297
LOCK ADMIN (Lock Out an Administrator)	303
LOCK NODE (Lock Out a Client Node)	304

MACRO (Invoke a Macro)	306
MOVE DATA (Move Files on a Storage Pool Volume)	309
QUERY ACTLOG (Search Activity Log for Messages)	311
QUERY ADMIN (Display Information on One or More Administrators)	317
QUERY ASSOCIATION (Query Client Node Associations with a Schedule)	322
QUERY AUDITOCCUPANCY (Query Client Node Storage Utilization)	325
QUERY CONTENT (Query the Contents of a Storage Pool Volume)	328
QUERY COPYGROUP (Query One or More Copy Groups)	334
QUERY DB (Display Information on the Database)	339
QUERY DBBACKUPTRIGGER (Query the Database Backup Trigger)	345
QUERY DBVOLUME (Display Information on One or More Database Volumes)	347
QUERY DEVCLASS (Display Information on One or More Device Classes)	350
QUERY DOMAIN (Query One or More Policy Domains)	354
QUERY DRIVE (Display Information about a Drive)	359
QUERY EVENT (Query Scheduled and Completed Events)	362
Querying Events for Client Schedules	362
Querying Events for Administrative Command Schedules	367
QUERY FILESPACE (Query One or More File Spaces)	377
QUERY LIBRARY (Query a Library)	381
QUERY LIBVOLUME (Query a Library Volume)	384
QUERY LICENSE (Display License Information)	386
QUERY LOG (Display Information on the Recovery Log)	389
QUERY LOGVOLUME (Display Information on One or More Log Volumes)	394
QUERY MGMTCLASS (Query a Management Class)	397
QUERY MOUNT (Display Information on Mounted Sequential Access Volumes)	401
QUERY NODE (Query One or More Client Nodes)	403
QUERY OCCUPANCY (Display File Space Information by Storage Pool)	408
QUERY OPTION (Query One or More Server Options)	412
QUERY POLICYSET (Query a Policy Set)	415
QUERY PROCESS (Query One or More Server Processes)	419
QUERY REQUEST (Query One or More Pending Mount Requests)	421
QUERY SCHEDULE (Query One or More Schedules)	422
QUERY SESSION (Query One or More Client Sessions)	427
QUERY STATUS (Query System Parameters)	431
QUERY STGPOOL (Query One or More Storage Pools)	436
QUERY VOLHISTORY (Display Sequential Volume History Information)	443
QUERY VOLUME (Query One or More Storage Pool Volumes)	450
QUIT (End the Interactive Mode of the Administrative Client)	456
REDUCE DB (Decrease the Assigned Capacity of the Database)	457
REDUCE LOG (Decrease the Assigned Capacity of the Recovery Log)	459
REGISTER ADMIN (Register an Administrator)	461
REGISTER LICENSE (Register a New License)	463
REGISTER NODE (Register a Client Node)	466
REMOVE ADMIN (Delete an Administrator)	470
REMOVE NODE (Delete a Client Node)	472
RENAME ADMIN (Rename an Administrator)	474
RENAME FILESPACE (Rename a Client Filespace on the Server)	476
RENAME NODE (Rename a Client Node)	479

REPLY (Allow a Request to Continue Processing)	481
RESET BUFPOOL (Reset the Buffer Pool Statistics for the Database)	482
RESET DBMAXUTILIZATION (Reset the Utilization Statistic for the Database)	484
RESET LOGCONSUMPTION (Reset the Cumulative Log Consumption Statistic)	485
RESET LOGMAXUTILIZATION (Reset the Utilization Statistic for the Log)	487
RESTORE STGPOOL (Restore Storage Pool Data from a Copy Pool)	488
RESTORE VOLUME (Restore Primary Volume Data from a Copy Pool)	493
REVOKE AUTHORITY (Remove Administrator Authority)	498
ROLLBACK (Rollback Uncommitted Changes in a Macro)	503
SET ACCOUNTING (Set Accounting Records On or Off)	505
SET ACTLOGRETENTION (Set the Retention Period for the Activity Log)	507
SET AUTHENTICATION (Set Password Authentication)	509
SET EVENTRETENTION (Specify the Retention Period for Event Records)	510
SET LICENSEAUDITPERIOD (Set License Audit Period)	512
SET LOGMODE (Set Mode for Saving Recovery Log Records)	514
SET MAXCMDRETRIES (Set Number of Times Scheduler Retries Commands)	517
SET MAXSCHEDULESESSIONS (Set Maximum Scheduled Sessions)	519
SET PASSEXP (Set Password Expiration Date)	521
SET QUERYSCHEDPERIOD (Set Query Period for Polling Client Nodes)	524
SET RANDOMIZE (Randomly Distribute Scheduled Start Times)	526
SET REGISTRATION (Set Open or Closed Registration)	528
SET RETRYPERIOD (Set Time between Retry Attempts)	530
SET SCHEDMODES (Select a Central Scheduling Mode)	532
SET SERVERNAME (Specify the Server Name)	534
UNLOCK ADMIN (Unlock an Administrator)	535
UNLOCK NODE (Unlock a Client Node)	536
UPDATE ADMIN (Update an Administrator)	537
UPDATE COPYGROUP (Update a Copy Group)	539
UPDATE DBBACKUPTRIGGER (Update the Database Backup Trigger)	547
UPDATE DEVCLASS (Change the Attributes of a Device Class)	550
Syntax for Updating a Device Class with a Device Type of 4MM	551
Syntax for Updating a Device Class with a Device Type of 8MM	554
Syntax for Updating a Device Class with a Device Type of DLT	557
Syntax for Updating a Device Class with a Device Type of FILE	560
Syntax for Updating a Device Class with a Device Type of LAN3995	561
Syntax for Updating a Device Class with a Device Type of MO35	562
Syntax for Updating a Device Class with a Device Type of OPTICAL	565
Syntax for Updating a Device Class with a Device Type of QIC	567
Syntax for Updating a Device Class with a Device Type of REMOVABLEFILE	570
UPDATE DOMAIN (Update a Policy Domain)	573
UPDATE DRIVE (Update a Drive)	575
UPDATE LIBRARY (Update a Library)	577
UPDATE LIBVOLUME (Change the Status of a Storage Volume)	579
UPDATE MGMTCLASS (Update a Management Class)	581
UPDATE NODE (Update a Client Node)	585
UPDATE POLICYSET (Update a Policy Set)	588
UPDATE SCHEDULE (Update a Client or an Administrative Command Schedule)	590
Updating Client Schedules	590

Updating Administrative Command Schedules . . . . .	597
UPDATE STGPOOL (Change the Attributes of a Storage Pool) . . . . .	605
UPDATE VOLHISTORY (Add the Location of a Database Dump, Database Backup, or Export Volume) . . . . .	617
UPDATE VOLUME (Change the Attributes of One or More Storage Pool Volumes) . . . . .	619
VALIDATE POLICYSET (Verify a Policy Set) . . . . .	625
VARY (Bring a Random Access Volume Online or Offline) . . . . .	629
<b>Appendix A. Miscellaneous Utilities . . . . .</b>	<b>633</b>
DSMFMT (Formatting Files for the Server) . . . . .	634
DSMLABEL (Labeling Volumes) . . . . .	637
GETIPXAD (Get the IPX Address) . . . . .	639
UPGRADEDDB (Upgrading from Version 1 to Version 2) . . . . .	640
<b>Appendix B. Recovery Log Extension Command . . . . .</b>	<b>641</b>
DSMSERV EXTEND LOG (Emergency Log Extension) . . . . .	642
<b>Appendix C. Database Recovery Commands . . . . .</b>	<b>645</b>
DSMSERV DISPLAY DBBACKUPVOLUME (Display Database Backup Volume Information) . . . . .	646
DSMSERV DISPLAY DBVOLUMES (Display Database Volume Information) . . . . .	649
DSMSERV DISPLAY LOGVOLUMES (Display Recovery Log Volume Information) . . . . .	651
DSMSERV RESTORE DB (Restore the Database) . . . . .	653
<b>Appendix D. Database Salvage Commands . . . . .</b>	<b>661</b>
DSMSERV DUMPDB (Dump the Database) . . . . .	662
DSMSERV INSTALL (Install the Database and Log) . . . . .	667
DSMSERV LOADDB (Reload the Database) . . . . .	669
DSMSERV AUDITDB (Verify Database Information) . . . . .	673
<b>Appendix E. Tailoring Server Options . . . . .</b>	<b>677</b>
Modifying the Server Options File . . . . .	677
Setting Communication Options . . . . .	677
Setting Client-Server Options . . . . .	678
Setting Date, Number, Time, and Language Options . . . . .	679
Setting Database and Recovery Log Options . . . . .	679
Setting Message Options . . . . .	680
Requesting Information about Server Options . . . . .	681
BUFPoolsize . . . . .	681
COMMOpendtimeout . . . . .	681
COMMTimeout . . . . .	682
DATEformat . . . . .	682
DEVCONFig . . . . .	683
EXPINterval . . . . .	683
EXPQUiet . . . . .	684
IDLETimeout . . . . .	685
IPXBufferSize . . . . .	685
IPXSocket . . . . .	685



LANAdapter	686
LANGuage	686
LOGPoolsize	687
LUname	687
MAXSessions	687
MSGSuppress	688
MESSageformat	688
MIRRORRead	689
MIRRORWrite	690
MOVEBatchsize	690
MOVESizethresh	691
NAMEdpipeName	691
NETBIOSBufferSize	692
NETBiosname	692
NETBIOSSession	693
NOAUDITStorage	693
NUMBERformat	693
SHMQUEUENAME	694
SNABufferSize	694
STATUSmsgcnt	695
TCPNodelay	695
TCPPort	696
TCPWindowSize	696
TIMEformat	696
TXNGroupmax	697
VOLUMEHISTORY	698
<b>Glossary</b>	<b>699</b>
<b>Index</b>	<b>717</b>



---

## Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, New York 10594, U.S.A.

Refer to the HONE SALESMANUAL or product announcement letters for the most current product information.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Information Enabling Requests  
Dept. M13  
5600 Cottle Road  
San Jose, CA 95193, U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM	DFSMS	Proprinter
AD/Cycle	DFSMS/MVS	PS/2
ADSTAR	DFSMSdss	RACF
Advanced Peer-to-Peer Networking	ESCON	RISC System/6000
AIX	IBM	RS/6000
AIX/6000	IBMLink	SAA
AIXwindows	Language Environment	SP2
Application System/400	Library Reader	System/370
APPN	MVS/DFP	System/390
AS/400	MVS/ESA	Systems Application Architecture
AT	MVS/SP	SystemView
BookManager	MVS/XA	Virtual Machine/Enterprise Systems Architecture
C/370	OpenEdition	Virtual Machine/Extended Architecture
CICS	Operating System/2	VM/ESA
Common User Access	Operating System/400	VM/XA
CUA	OS/2	VSE/ESA
DATABASE 2	OS/400	VTAM
DB2/6000	POWERparallel	WIN-OS/2

The following terms are trademarks of other companies:

<b>Trademark</b>	<b>Company</b>	<b>Trademark</b>	<b>Company</b>
Andataco	Andataco Corporation	NetWare	Novell, Inc.
Apple	Apple Computer, Inc.	NFS	Sun Microsystems, Inc.
Attachmate	Attachmate Corporation	Novell	Novell, Inc.
CompuServe	CompuServe, Inc.	Open Desktop	The Santa Cruz Operation, Inc.
dBASE	Borland International, Inc.	OpenWindows	Sun Microsystems, Inc.
DECstation	Digital Equipment Corporation	PARADOX	Borland International, Inc.
DLT	Quantum Corporation	PC/TCP	FTP Software, Inc.
DPX/20	Groupe Bull	PTX	Sequent Computer Systems
Dynatek	Dynatek Automation Systems	SCO	The Santa Cruz Operation, Inc.
DynaText	Electronic Book Technologies, Inc.	Sequent	Sequent Computer Systems
Exabyte	Exabyte Corporation	SINIX	Siemens Nixdorf Information Systems, Inc.
Extra!	Attachmate Corporation		
FOXPRO	Microsoft Corporation	Solaris	Sun Microsystems, Inc.
Hewlett-Packard	Hewlett-Packard Company	Sony	Sony Corporation
HP-UX	Hewlett-Packard Company	SPARC	SPARC International, Inc.
Ice Box	Software International Microsystems	StorageTek	Storage Technology Corporation
iFOR/LS	Gradient Technologies, Inc.	Sun	Sun Microsystems, Inc.
INGRES	ASK Group, Inc.	Sun Microsystems	Sun Microsystems, Inc.
Intel	Intel Corporation	SunOS	Sun Microsystems, Inc.
lomega	lomega Corporation	Sun-3	Sun Microsystems, Inc.
IPX/SPX	Novell, Inc.	Sun-4	Sun Microsystems, Inc.
IRIX	Silicon Graphics, Inc.	SureStore	Hewlett-Packard Company
Jetstore	Hewlett-Packard Company	SyQuest	SyQuest Technology, Inc.
Lotus	Lotus Development Corporation	ULTRIX	Digital Equipment Corporation
Lotus Notes	Lotus Development Corporation	WangDAT	Tecmar Technologies, Inc.
Macintosh	Apple Computer, Inc.	Windows 95	Microsoft Corporation
MacTCP	Apple Computer, Inc.	Windows NT	Microsoft Corporation
Motif	Open Software Foundation, Inc.	X Windows	Massachusetts Institute of Technology

C-bus is a trademark of Corollary, Inc.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

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

Other company, product, and service names, which may be denoted by a double asterisk (\*\*), may be trademarks or service marks of others.



---

## Preface

ADSTAR Distributed Storage Manager (ADSM) is a client/server program that provides storage management to customers in a multivendor computer environment.

ADSM provides an automated, centrally scheduled, policy-managed backup, archive, and space management facility for file-servers and workstations.

This publication provides reference information for administrators of ADSM.

You can define and manage ADSM processes through administrator commands or a graphical user interface.

---

### Who Should Read This Publication

This reference is intended for anyone who has been registered as an administrator. While ADSM can be managed by a single administrator, administrative responsibilities can be divided among a number of people as an installation requires.

All of the administrator commands you need to operate and maintain ADSM can be invoked either from the server console or by using an administrative client running on a workstation connected to the server.

---

### What You Should Know Before Reading This Publication

You should be familiar with the operating system on which the server resides and the communication protocols required for the client/server environment.

For information on product requirements for ADSM, see *ADSM Licensed Program Specifications*.

You also need to understand the storage management practices of your organization, such as how you are currently backing up your workstation files and how you are using random access media and sequential access media.

For information on installing ADSM, see *ADSM Quick Start*.

---

### ADSTAR Distributed Storage Manager Publications

The following table lists ADSM publications. For a complete list, see *ADSM General Information*.

The ADSM library is also available in softcopy on the *ADSM Online Product Library* CD-ROM (order number SK2T-1893) and the *IBM Online Library Omnibus Edition: AIX Collection* CD-ROM (order number SK2T-2066).

<b>Short Title</b>	<b>Publication Title</b>	<b>Order Number</b>
ADSM General Information	<i>ADSTAR Distributed Storage Manager: General Information</i>	GH35-0131
ADSM Messages	<i>ADSTAR Distributed Storage Manager: Messages</i>	SH35-0133
ADSM License Information	<i>ADSTAR Distributed Storage Manager for OS/2: License Information</i>	GC35-0230
ADSM Quick Start	<i>ADSTAR Distributed Storage Manager for OS/2: Quick Start</i>	GC35-0231
ADSM Administrator's Guide	<i>ADSTAR Distributed Storage Manager for OS/2: Administrator's Guide</i>	GC35-0232
ADSM Using the UNIX HSM Clients	<i>ADSTAR Distributed Storage Manager: Using the UNIX Hierarchical Storage Management Clients</i>	SH26-4030
ADSM V2 Using the Apple Macintosh Backup-Archive Client	<i>ADSTAR Distributed Storage Manager Version 2: Using the Apple Macintosh Backup-Archive Client</i>	SH26-4051
ADSM Using the UNIX Backup-Archive Clients	<i>ADSTAR Distributed Storage Manager Version 2: Using the UNIX Backup-Archive Clients</i>	SH26-4052
ADSM V2 Using the OS/2 Backup-Archive Client	<i>ADSTAR Distributed Storage Manager Version 2: Using the OS/2 Backup-Archive Client</i>	SH26-4053
ADSM V2 Using the DOS Backup-Archive Client	<i>ADSTAR Distributed Storage Manager Version 2: Using the DOS Backup-Archive Client</i>	SH26-4054
ADSM V2 Using the Novell NetWare Backup-Archive Client	<i>ADSTAR Distributed Storage Manager Version 2: Using the Novell NetWare Backup-Archive Client</i>	SH26-4055
ADSM V2 Using the Microsoft Windows Backup-Archive Clients	<i>ADSTAR Distributed Storage Manager Version 2: Using the Microsoft Windows Backup-Archive Clients</i>	SH26-4056
ADSM Using the Lotus Notes Backup Agent	<i>ADSTAR Distributed Storage Manager: Using the Lotus Notes Backup Agent</i>	SH26-4047
ADSM Installing the Clients	<i>ADSTAR Distributed Storage Manager: Installing the Clients</i>	SH26-4049
ADSM Client Reference Cards	<i>ADSTAR Distributed Storage Manager: Client Reference Cards</i>	SX26-6013

## **Software Developer's Program**

The IBM Storage Systems Division (SSD) Software Developer's Program provides a range of services to software developers who want to use the ADSM application



programming interface (API). Information about the SSD Software Developer's Program is available in:

- IBMSTORAGE forum on CompuServe
- SSD Software Developer's Program Information Package

To obtain the Software Developer's Program Information Package:

1. Call 800-4-IBMSSD (800-442-6773). Outside the U.S.A., call 408-256-0000.
2. Listen for the Storage Systems Division Software Developer's Program prompt.
3. Request the Software Developer's Program Information Package.

---

## Translations

Selected ADSM publications have been translated into languages other than American English. For a complete list of the available translations and their order numbers, see *ADSM General Information*. Contact your IBM representative for more information about the translated publications and whether these translations are available in your country.

---

## Do You Have Comments or Suggestions?

If you have difficulty using this publication or if you have comments and suggestions for improving it, please complete and mail the readers' comment form found in the back of this publication. Your comments and suggestions can contribute to the quality and usability of this publication.



---

## Changes for ADSTAR Distributed Storage Manager Version 2

The new functions for ADSM Version 2 are:

### **Administrative command scheduling**

You can define schedules for automatically issuing administrative commands once or periodically.

See the following changed commands for details:

```
DEFINE SCHEDULE
COPY SCHEDULE
QUERY EVENT
QUERY SCHEDULE
UPDATE SCHEDULE
```

### **Configuration and administration enhancements**

You can use the ADSM utilities interface to make ADSM configuration and administration tasks easier.

### **Database backup and recovery**

You can perform full and incremental backups of the server database to protect against loss or damage. You can use the backup copies to restore the database to its current state or to a specific point in time. You can back up the database while the server is available to clients.

**Note:** To allow for recovery of the database to its most current state, you may have to extend your recovery log space significantly.

See the following new commands for details:

```
BACKUP DB
BACKUP DEVCONFIG
DEFINE DBBACKUPTRIGGER
DELETE DBBACKUPTRIGGER
QUERY DBBACKUPTRIGGER
RESET LOGCONSUMPTION
SET LOGMODE
UPDATE DBBACKUPTRIGGER
UPDATE VOLHISTORY
```

See also the following new offline commands for details:

```
DSMSERV DISPLAY DBBACKUPVOLUMES
DSMSERV DISPLAY DBVOLUMES
DSMSERV DISPLAY LOGVOLUMES
DSMSERV RESTORE DB
```

Two options, VOLUMEHISTORY and DEVCONFIG, can be added to the server options file to back up sequential volume history information and device configuration information to files that can be used during recovery.

The online DUMP DB command is removed.

### **Device support enhancements**

Device class types now include support for 4mm tape drives, quarter-inch cartridge (QIC) tape drives, DLT tape drives, and any drive that supports removable file systems. Library device support now allows the following:

- The user can select whether media labels are read when volumes are checked in and checked out.
- ADSM can initiate a swap operation when an empty library slot is not available during check-in processing.

ADSM can operate in conjunction with external media management systems.

See the following changed commands for details:

```
DEFINE DEVCLASS  
DEFINE LIBRARY  
CHECKIN LIBVOLUME  
CHECKOUT LIBVOLUME  
UPDATE DEVCLASS  
UPDATE LIBRARY
```

### **Hierarchical storage management**

Hierarchical storage management (HSM) provides space management services to HSM clients. HSM clients can automatically migrate user files to storage pools to free up client storage space. A user can access a migrated file as if it were on local storage.

See the following changed commands for details:

```
DEFINE MGMTCLASS  
UPDATE MGMTCLASS
```

### **Storage pool backup and recovery**

You can back up client files stored on storage pools to sequential media. These media can be either onsite, to protect against media loss, or offsite, for disaster recovery purposes.

See the following new commands for details:

```
BACKUP STGPOOL  
RESTORE STGPOOL  
RESTORE VOLUME
```

See also the following changed commands:

```
DEFINE STGPOOL  
DEFINE VOLUME  
UPDATE STGPOOL  
UPDATE VOLUME
```

A new shared memory communication protocol is also supported.

SHAREDMEMORY is for use when client and server are on the same system. SHAREDMEMORY uses less overhead than Named Pipes.

## Chapter 1. Using Syntax Diagrams

This section describes how to read the syntax diagrams included in the following sections for each administrator command.

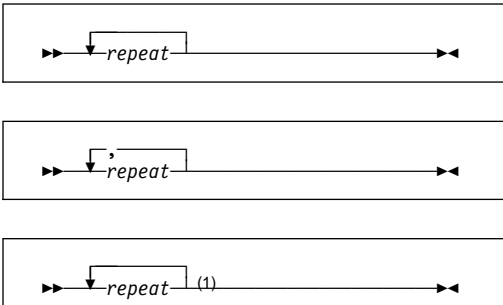
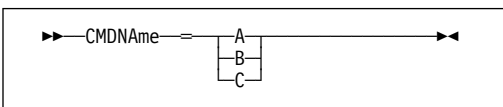
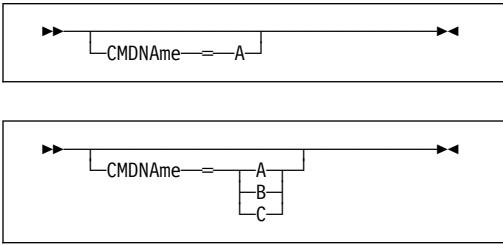
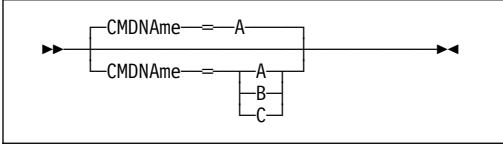
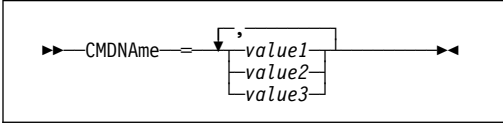
To read a syntax diagram, follow the path of the line. Read from left to right and top to bottom.

- The **▶▶—** symbol indicates the beginning of a syntax diagram.
- The **—▶▶** symbol, at the end of a line, indicates that the syntax diagram continues onto the next line.
- The **▶—** symbol, at the beginning of a line, indicates that a syntax diagram continues from the previous line.
- The **—▶▶** symbol indicates the end of a syntax diagram.

Syntax items (for example, a keyword or variable) may be:

- Directly on the line (required)
- Above the line (default)
- Below the line (optional)

Syntax Diagram Description	Example
<p><b>Abbreviations:</b> Uppercase letters denote the shortest acceptable abbreviation. If an item appears entirely in uppercase letters, it cannot be abbreviated.</p> <p>You can type the item in uppercase letters, lowercase letters, or any combination.</p> <p>In this example, you can enter CMDNA, CMDNAM, or CMDNAME in any combination of uppercase and lowercase letters.</p>	
<p><b>Symbols:</b> You must code these symbols exactly as they appear in the syntax diagram.</p>	<ul style="list-style-type: none"> <li>* Asterisk</li> <li>:</li> <li>,</li> <li>= Equal Sign</li> <li>- Hyphen</li> <li>() Parentheses</li> <li>.</li> </ul>
<p><b>Variables:</b> Highlighted lowercase items (<i>like this</i>) denote variables.</p> <p>In this example, <i>var_name</i> represents a variable you must specify when you code the CMDNAME command.</p>	

Syntax Diagram Description	Example
<p><b>Repetition:</b> An arrow returning to the left means that the item can be repeated.</p> <p>A character within the arrow means you must separate repeated items with that character.</p> <p>A footnote (1) by the arrow refers to a limit that tells how many times the item can be repeated.</p>	 <p><b>Note:</b> 1 Specify <i>repeat</i> up to 5 times.</p>
<p><b>Required Choices:</b> When two or more items are in a stack and one of them is on the line, you <i>must</i> specify one item.</p> <p>In this example, you must choose <code>CMDNAME=A</code>, <code>CMDNAME=B</code>, or <code>CMDNAME=C</code>. Do not include any blanks immediately before or after the equal sign (=).</p>	
<p><b>Optional Choice:</b> When an item is below the line, the item is optional. In this example, you can choose <code>CMDNAME=A</code> or nothing at all. Do not include any blanks immediately before or after the equal sign (=).</p> <p>When two or more items are in a stack below the line, all of them are optional. In this example, you can choose <code>CMDNAME=A</code>, <code>CMDNAME=B</code>, <code>CMDNAME=C</code>, or nothing at all. Do not include any blanks immediately before or after the equal sign (=).</p>	
<p><b>Defaults:</b> Defaults are above the line. The system uses the default unless you override it. You can override the default by coding an option from the stack below the line.</p> <p>In this example, <code>CMDNAME=A</code> is the default. You can also code <code>CMDNAME=A</code>, <code>CMDNAME=B</code> or <code>CMDNAME=C</code>. Do not include any blanks before or after the equal sign (=).</p>	
<p><b>Repeatable Choices:</b> A stack of items followed by an arrow returning to the left means that you can select more than one item or, in some cases, repeat a single item.</p> <p>In this example, you can choose more than one value, with each name delimited with a comma. Do not include any blanks before or after the equal sign (=).</p>	

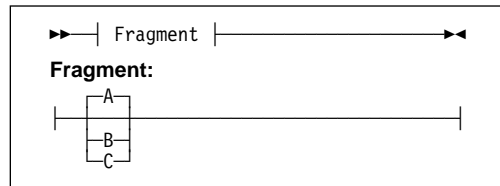
---

**Syntax Diagram Description****Syntax Fragments:**

Some diagrams, because of their length, must fragment the syntax. The fragment name appears between vertical bars in the diagram. The expanded fragment appears in the diagram after all other parameters or at the bottom of the diagram. The expanded fragment is identified by a heading with the fragment name. Commands appearing directly on the line are required.

In this example, the fragment is named "Fragment."

---

**Example**





---

## Chapter 2. Entering Administrative Commands

This section includes the following information about entering ADSTAR Distributed Storage Manager (ADSM) commands:

- Descriptions of the components of a command
- Rules for entering administrative commands
- How ADSM processes administrative commands
- How ADSM formats command output
- Instructions for using wildcard characters
- Naming conventions for ADSM objects

---

### Command Components

An administrative command can include one or more of these attributes:

- Command name
- Positional parameters
- Keyword parameters

The attributes of administrative commands are also discussed in Chapter 1, "Using Syntax Diagrams" on page 1.

### Command Name

The first component of a command is the command name. The command name is the unique identifier for the command. It can consist of a single action word, such as HALT or ENABLE, or it can consist of an action word and an object for the action, such as DEFINE DOMAIN.

You can enter the entire command name or the abbreviation specified in the syntax diagram for the command.

### Positional Parameters

Some commands have required or optional positional parameters. Positional parameters must be entered in the order indicated in the syntax diagram for a command.

For example, to copy a policy set, enter the name of the policy domain to which the policy set belongs, the name of the policy set you want to copy, and the name you want to assign to the copy, in that order:

```
COPY Policyset domain1 oldset newset
```

### Keyword Parameters

Some commands have required or optional keyword parameters. When included in a command, a keyword parameter must follow any positional parameters in the command. A keyword parameter consists of a keyword, followed by an equal sign, followed by a value. Do not include any blanks immediately before or after the equal

sign (=). A keyword parameter can, for example, provide a description of an object, identify an object (such as a storage pool or policy domain), or specify a setting for an option.

For example, to include a description for a new policy domain, enter:

```
DEFine D0main acctg DESCription="Accounting department"
```

You can enter either the entire keyword, or the abbreviation identified in the syntax diagram for the command.

---

## Command Entry Rules

The following are general rules for entering ADSM administrative commands:

- Enter a command in any column of the input line.
- Unless otherwise noted, commands can be specified in any combination of uppercase and lowercase characters.
- If a parameter allows a list of values, enter the values immediately following the equal sign and delimit the value with a comma. For example,
 

```
CLasses=poli cy, storage
```
- If a description value (a string of text) for a parameter begins with a single or double quote or contains any embedded blanks or equal signs, you must surround the value in either single (') or double (") quotation marks. The opening and closing quotation marks must be the same type of quotation marks. For example, if the opening quotation is a single quotation mark, the closing quotation mark must also be a single quotation mark. See the following table for details on how to enter description values.

For example, to register a new client node with a description for the CONTACT parameter, enter:

```
REGister Node louie2 secret CONtact="manager"
```

The following table presents ways of entering a description value for the CONTACT parameter, using the REGISTER NODE command as an example. The value can contain quotation marks, embedded blanks, or equal signs.

For this description	Enter this
manager	CONtact=manager
manager's	CONtact="manager's" <b>or</b> CONtact='manager' 's'
manager's data	CONtact="manager's data" <b>or</b> CONtact='manager' 's data'
manager's "data"	CONtact='manager' 's "data" '
"manager"	CONtact=' "manager" ' <b>or</b> CONtact=" "manager" ""
manager=2GF	CONtact='manager=2GF'

<b>For this description</b>	<b>Enter this</b>
manager reporting to 2GF	CONtact='manager reporting to 2GF' <b>or</b> CONtact="manager reporting to 2GF"

---

## **Processing Administrative Commands**

ADSM processes administrator commands in one of two ways:

- Foreground
- Background

ADSM sends you a message indicating:

- Whether the command completed successfully (in the case of a foreground process)
- Whether the command has been sent to the server to be processed (in the case of a background process)
- A confirmation message, in the case of commands that affect the availability of the server or data managed by the server. (Unless the command is issued from the server console or from an administrative client with the NOCONFIRM option.)

### **Commands Processed in the Foreground**

Foreground processing means that each command you issue must complete before you can issue another command. For example, when you enter a query command, the information you request is sent to your screen. You cannot issue another command until this happens. It is possible for additional commands to be issued from different administrative clients while your command is processed.

You cannot cancel commands that are processed in the foreground. When you issue a QUERY command from the administrative client, multiple screens of output may be generated. If this occurs and additional output is not needed, the administrator can cancel the display of output to the client workstation. While canceling does not end the processing of the command, it ends any further display of command output.

### **Commands Processed in the Background**

Background processing permits you to enter commands while the server is processing a previously issued command.

When you enter a command that creates a background process, ADSM submits a request to the server to perform the task identified in the command. The following is a list of commands that can generate background processes:

AUDIT LIBRARY	EXPORT NODE
AUDIT LICENSE	EXPORT POLICY
AUDIT VOLUME	EXPORT SERVER
BACKUP DB	EXTEND LOG
BACKUP STGPOOL	IMPORT ADMIN
CHECKIN LIBVOLUME	IMPORT NODE
CHECKOUT LIBVOLUME	IMPORT POLICY
DEFINE DBCOPY	IMPORT SERVER
DEFINE LOGCOPY	MOVE DATA
DELETE DBVOLUME	REDUCE DB
DELETE FILESPACE	REDUCE LOG
DELETE LOGVOLUME	RESTORE STGPOOL
DELETE VOLUME	RESTORE VOLUME
EXPIRE INVENTORY	VARY
EXPORT ADMIN	

Each background process is assigned a process number. You can cancel commands that generate background processes using this process number.

Table 1 identifies the commands you use to obtain the status and process number of a background process and how to cancel a background process. If a background process is active when you cancel it, the server stops the process. Any changes that are uncommitted are rolled back, however, changes that are committed are not rolled back.

*Table 1. Querying and Canceling a Process*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## Formatting Command Output on a Terminal

ADSM formats command output according to your screen or window width. If the width of your screen or window is not wide enough to display the output horizontally, ADSM arranges and displays the information vertically.

## Using Wildcard Characters

In some commands, such as the query commands, you can use wildcard characters to create a pattern-matching expression that specifies more than one object. Using wildcard characters makes it easier to tailor a command to your needs.

The wildcard characters you use depend on the operating system from which you issue commands. For example, you can use wildcard characters such as an asterisk (\*) to match any (0 or more) characters or you can use a question mark (?), or a percent sign (%) to match exactly one character.

Table 2 provides references to wildcard characters for some operating systems. Use wildcard characters appropriate for your system.

*Table 2. Wildcard Characters by Environment*

<b>Environment</b>	<b>Match-Any</b>	<b>Match-Exactly-One</b>
AIX/UNIX	*	?
HP	*	%
AS/400	*	%
CMS	*	%
DOS	*	?
OS/2	*	?
SunOS	*	?
TSO	*	%
Windows	*	?

For example, if you want to query all the management classes whose names begin with DEV in all the policy sets in DOMAIN1, and your system uses an asterisk as the *match-any* character, you can enter:

```
Query Mgmtclass domain1 * dev*
```

Or, if you want to query the first nine management classes in POLICYSET1 in DOMAIN1, and POLICYSET1 contains twelve management classes named MC1 through MC12, and your system uses a question mark as the match-one character, you can enter:

```
Query Mgmtclass domain1 policyset1 mc?
```

ADSM displays information about management classes named MC1 through MC9.

## Examples for Match-Any Characters

Table 3 shows additional examples using wildcard characters to match any characters.

*Table 3. Match-Any Character*

<b>Pattern</b>	<b>Matches</b>	<b>Does Not Match</b>
ab*	ab, abb, abxxx	a, b, aa, bb
ab*rs	abrs, abtrs, abrsrs	ars, aabrs, abrss
ab*ef*rs	abefrs, abefghrs	abeft, abers

## Examples for Match-Exactly-One Character

Table 4 shows additional examples using wildcard characters to match exactly one character. The question mark (?) can be replaced by a percent sign (%) if your platform uses that character instead of ?.

Table 4. Match-Exactly-One Character

Pattern	Matches	Does Not Match
ab?	abc	ab, abab, abzzzz
ab?rs	abfrs	abrs, abllrs
ab?ef?rs	abdefjrs	abefrs, abdefrs, abefjrs
ab??rs	abcdrs, abzzrs	abrs, abjrs, abkkrs

---

## Naming Conventions

This section provides information about naming conventions for ADSM objects.

### Naming Volumes

When you use DEFINE commands to define volumes for the database, recovery log and storage pools, the volume name is dependent on the type of sequential access media or random access media you are using. This publication provides the command syntax and rules to follow when naming these volumes. For a detailed discussion on setting up these volumes, see *ADSM Administrator's Guide*.

### Case-Sensitivity

When referring to file space names, you must specify the file space name using the exact capitalization that is used by ADSM. Use the QUERY FILESPACE command to find out the correct capitalization.

## Naming Restrictions for Objects

ADSM restricts the number and type of characters that you can use to name ADSM objects.

**Note:** ADSM object names cannot contain any embedded blanks or equal signs.

### Maximum Length

The following list shows the maximum length permitted for naming objects in ADSM:

Type of Name	Maximum Length
Administrators	64
Device classes	30
Management classes	30
Nodes	64
Passwords	64
Policy domains	30
Policy sets	30
Schedules	30
Server name	64
Storage pools	30

### Valid Characters

The following characters are available to you for defining ADSM object names:

Characters	Character Names
A–Z	Any letter, A through Z
0–9	Any number, 0 through 9
_	Underscore
.	Period
-	Hyphen
+	Plus
&	Ampersand





---

## Chapter 3. Using the Command Line Interface

This section provides information about:

- Using the server console session
- Using the administrative client session
- Using administrative client options
- Using continuation characters to issue long commands
- Redirecting command output to a file
- Using macros to issue commands

---

### Using the Server Console Session

ADSM has set up a special administrative user ID named `SERVER_CONSOLE` that allows you to administer the server from the server console after ADSM is installed. At installation, `SERVER_CONSOLE` is automatically registered as an administrator and is given system authority. You can use the `SERVER_CONSOLE` administrative ID to issue ADSM administrative commands.

Because the `SERVER_CONSOLE` administrative ID is provided with ADSM, there are certain functions you cannot complete against this user ID. These include:

- Registering or updating `SERVER_CONSOLE`
- Locking or unlocking `SERVER_CONSOLE` from ADSM
- Renaming `SERVER_CONSOLE`
- Removing `SERVER_CONSOLE`

The `SERVER_CONSOLE` administrative ID does *not* receive a confirmation message when issuing commands that affect the availability of the server or data managed by the server. For more information on how to issue commands from the `SERVER_CONSOLE` administrative ID, see the *ADSM Quick Start*.

---

### Using the Administrative Client Session

The administrative command-line client is a program that allows administrators to control and monitor the server from a remote workstation rather than from the server console.

The administrative client program can be installed on a workstation, personal computer, or terminal running in one of the following environments:

- Digital Equipment Corporation (DECstation)
- Hewlett Packard HP-UX for HP 9000, Series 700 and Series 800
- IBM Advanced Interactive Executive (AIX) for the RISC System/6000
- IBM Disk Operating System (DOS)
- IBM MVS Time Sharing Option (TSO)
- IBM Operating System/2 (OS/2)
- IBM VM CMS
- Microsoft Windows

- Microsoft Windows NT
- Santa Cruz Operation (SCO)
- Sun Microsystems SunOS

You can start an administrative client session using any of the following modes:

- Console
- Batch
- Mount
- Interactive

## Using Console Mode

Use console mode for your administrative client session when you want to monitor ADSM activities. In console mode, you can monitor basic server activities as they occur. For example, you can monitor migration processes and clients logging on to ADSM. This information is displayed on your terminal, and optionally, can be written to a file using the `OUTFILE` option.

You *cannot* enter any ADSM administrator commands while running in console mode.

### Starting an Administrative Client Session in Console Mode

To start an administrative client session in console mode:

1. Enter the `DSMADMC` command with the `CONSOLEMODE` option:

```
DSMADMC -CONSOLEMODE
```

2. Press the Enter key. The following prompt appears on your screen:

```
Enter your user id:
```

3. Enter your ADSM administrator ID, and then press the Enter key.

This prompt appears on your screen if authentication is turned on for the server.

```
Enter your password:
```

4. Enter your ADSM administrator password, and then press the Enter key. ADSM runs in console mode until you end your session.

If you want ADSM to write all terminal output to a file, specify the `OUTFILE` option with a destination file name. For example on OS/2, to have output written to the `SAVE.OUT` file, enter:

```
DSMADMC -CONSOLEMODE -OUTFILE=save.out
```

All ADSM activity messages are written to the `SAVE.OUT` file.

You can also enter your user ID and password in the `DSMADMC` command, using the `ID` and `PASSWORD` options, if you do not want to be prompted for that information. For example on OS/2, enter:

```
DSMADMC -ID=sullivan -PASSWORD=secret -CONSOLEMODE -OUTFILE=save.out
```

For more information on the ID, PASSWORD, CONSOLEMODE, and OUTFILE options, see “Using Administrative Client Options” on page 18.

### Ending an Administrative Client Session in Console Mode

To end an administrative client session in console mode, use one of the following keyboard break sequences:

Environment	Break Sequence
AIX and UNIX-based clients	Ctrl+C
CMS	HX
DOS	Ctrl+C, Ctrl+Break
OS/2	Ctrl+C, Ctrl+Break
SunOS	Ctrl+C
TSO	ATTN
Windows and Windows NT	Ctrl+C, Ctrl+Break

### Using Mount Mode

Use mount mode for your administrative client session when you want to monitor ADSM removable-media mount activities. This information is displayed on your terminal, and optionally, can be written to a file using the OUTFILE option.

You *cannot* enter any ADSM administrator commands while running in mount mode.

### Starting an Administrative Client Session in Mount Mode

To start an administrative client session in mount mode, for example on OS/2:

1. Enter the DSMADMC command with the MOUNTMODE option:

```
DSMADMC -MOUNTmode
```

2. Press the Enter key. The following prompt appears on your screen:

```
Enter your user id:
```

3. Enter your ADSM administrator ID, and then press the Enter key.

This prompt appears on your screen if authentication is turned on for the server.

```
Enter your password:
```

4. Enter your ADSM administrator password, and then press the Enter key. ADSM runs in mount mode until you end your session.

If you want ADSM to write all terminal output to a file, specify the OUTFILE option with a destination file name. For example on OS/2, to have output written to the SAVE.OUT file, enter:

```
DSMADMC -MOUNTmode -OUTFILE=save.out
```

All ADSM activity messages are written to the SAVE.OUT file.

You can also enter your user ID and password in the DSMADMC command, using the ID and PASSWORD options, if you do not want to be prompted for that information. For example, enter:

```
DSMADMC -ID=sullivan -PAssword=secret -MOUNTmode -OUTfile=save.out
```

For more information on the ID, PASSWORD, MOUNTMODE, and OUTFILE options, see “Using Administrative Client Options” on page 18.

### **Ending an Administrative Client Session in Mount Mode**

To end an administrative client session in mount mode, use the Ctrl+C keyboard break sequence.

## **Using Batch Mode**

Use batch mode for your administrative client session when you want to enter a single administrator command or a MACRO command. To start an administrative client session in batch mode, a server session must be available. Your administrative client session automatically ends when the command or macro has executed.

To start an administrative client session in batch mode, for example on OS/2, enter the DSMADMC command with your user ID and password, using the ID and PASSWORD options, respectively.

For example, to issue the ENABLE command in batch, enter:

```
DSMADMC -ID=sullivan -PAssword=secret ENABLE
```

You can enter any of the other DSMADMC command options (*except* CONSOLEMODE or MOUNTMODE) that you want to use.

If you specify a parameter with a string of text using batch mode, enclose the text in single quotation marks ( ' ') Double quotes are not allowed in batch mode.

In batch mode, you must enter the complete command on one line. If a command does not fit on one line and you still want to enter it in batch mode, you can enter the command by using a macro file. Also, if you want to use single quotation marks in batch mode, you must use them inside a macro file. For more information about using macro files, see “Using Macros to Issue Commands” on page 21.

For more information on the administrative client options, see “Using Administrative Client Options” on page 18.

## **Using Interactive Mode**

Use interactive mode for your administrative client session when you want to enter a series of administrative commands. To start an administrative client session in interactive mode, a server session must be available. To ensure the availability of server sessions for both administrative and client node sessions, the interactive mode

of the administrative client is disconnected if one or more of the following conditions is true:

- The server was stopped using the HALT command.
- The administrative client session did not issue any commands for the length of time (in minutes) specified in the server options file.
- The administrative client session was canceled with the CANCEL SESSION command.

You can automatically restart your administrative client session by entering another command.

### Starting an Administrative Client Session in Interactive Mode

To start your administrative client session in interactive mode, for example on OS/2:

1. Enter

```
DSMADMC
```

The following prompt appears on your screen:

```
Enter your user id:
```

2. Enter your ADSM administrator ID, and then press the Enter key. This prompt appears on your screen if authentication is turned on for the server.

```
Enter your password:
```

3. Enter your ADSM administrator password, and then press the Enter key. The following prompt appears on your screen:

```
adsm>
```

4. Enter an administrative command each time ADSM displays the **adsm>** prompt.

When command processing is complete, the **adsm>** will appear and the administrator can then perform additional work. Command processing can be stopped by either canceling the existing client session or stopping the client program.

You can use the ID and PASSWORD options to specify your ADSM administrator ID and password if you do not want to be prompted for this information. For example on OS/2, enter:

```
DSMADMC -ID=sullivan -PAssword=secret
```

In addition to the ID and PASSWORD options, you can use any of the administrative client options (except CONSOLEMODE or MOUNTMODE) with the interactive mode of

the administrative client. For more information on the administrative client options, see “Using Administrative Client Options” on page 18.

You can use continuation characters in the interactive mode of the administrative client. For more information, see “Using Continuation Characters” on page 19.

### **Ending an Administrative Client Session in Interactive Mode**

Use the QUIT command to end an administrative client session in interactive mode.

## **Using Administrative Client Options**

In all administrative client modes, the administrative client options modify your administrative client session responses.

The following administrative client options are valid from an administrative client session only:

**-ID=userid**

Specifies the ADSM administrator's user ID.

**-PAssword=password**

Specifies the ADSM administrator's password.

**-NOConfirm**

Specifies that you do not want ADSM to request confirmation before executing commands that affect the availability of the server or data managed by the server.

**-Quiet**

Specifies that you do not want ADSM to display standard output messages to your screen. However, when you use this option, certain error messages still appear.

**-CONsolemode**

Specifies that you want to run ADSM in console mode. All unsolicited server console output is echoed to your terminal. You cannot enter any administrator commands while ADSM is running in console mode.

**-MOUNTmode**

Specifies that you want to run ADSM in mount mode. All server removable-media mount messages are echoed to your terminal. You cannot enter any administrator commands while running in mount mode.

**-Itemcommit**

Specifies that you want ADSM to commit commands inside a macro as each command is executed. For more information, see “Controlling Command Processing in a Macro” on page 25.

**-OUTfile=filename**

Specifies that you want ADSM to write all terminal output to a specified file. In an interactive or console modes, the output goes to the terminal. In a batch mode, output is written to the specified file only.

Besides the options listed here, you can also specify any option that is in the client options file. Each option must be preceded with a hyphen and delimited with a space.

## Using Continuation Characters

You can use continuation characters in the interactive mode (loop) of the administrative client or in a macro file. Continuation characters are useful when you want to execute a command that is longer than your screen or window width. You can:

- Enter a dash or a back slash at the end of the line you want to continue.

For example:

```
REGister Admin pease mypasswd -  
CONtact="David, ext1234"
```

The maximum number of characters you can enter without a continuation character is 256. The maximum number of characters you can enter with continuation characters is 1500.

- If you want to continue a list of values, enter a dash or a back slash, with no preceding blank spaces, after the last comma of the list that you enter on the first line. Then, enter the remaining items in the list on the next line with no preceding blank spaces. For example:

```
STGpools=stg1,stg2,stg3,-  
stg4,stg5,stg6
```

- To continue a string of values that are enclosed in quotation marks, enter the first part of the string enclosed in quotation marks, followed by a dash or a back slash at the end of the line. Then, enter the remainder of the string on the next line enclosed in the *same* type of quotation marks.

For example:

```
CONtact="DAVID PEASE, BLDG. 100, ROOM 2B, SAN JOSE, "-  
"EXT. 1234, ALTERNATE CONTACT-NORM PASS, EXT. 2345"
```

ADSM concatenates the two strings with no intervening blanks. You must use *only* this method to continue a quoted string of values across more than one line.

## Redirecting Command Output to a File in Batch and Interactive Modes

You can save the output from a command by entering ADSM administrative client redirection characters at the end of the command. Redirection characters (> or >>) direct the output of a command to a file you specify instead of to your screen. Redirection is only available from the administrative client in the batch and interactive modes. Redirection can also be used in console mode and mount mode in environments that support redirection natively.

You can use console mode for your administrative client session to pipe command output into a program. This allows you to write your own programs to automate operations. For example:

```
dsmadmc -console -id=admin -pa=xxx | filter.exe
```

would pipe all output going to the administrative client in console mode to a program called filter.exe. The program then monitors the output for individual messages as they occur and takes appropriate action, such as sending mail to another user.

When redirecting output to a file, follow the naming conventions of the administrative client running on your operating system. For example, the naming conventions for some operating systems are presented below.

- On AIX- and UNIX-based operating systems, a file name is case-sensitive and can contain as many as 64 alphanumeric characters, with unlimited use of periods (.) for delimiting the file name.
- On DOS and Windows operating systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
- On a Windows NT 3.5 operating system:
  - For high-performance file system (HPFS), a file name is not case-sensitive and can contain as many as 255 characters.
  - For file allocation table (FAT) file systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
  - For NT file system (NTFS), a file name can contain as many as 255 characters (including spaces and periods).
- On a Windows NT 4.0 operating system:
  - For file allocation table (FAT) file systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
  - For NT file system (NTFS), a file name can contain as many as 255 characters (including spaces and periods).
- On an OS/2 operating system:
  - For file allocation table (FAT) file systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
  - For high-performance file system (HPFS), a file name is not case-sensitive and can contain as many as 255 characters.
- On a VM operating system, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an 8-character, alphanumeric file type. You can delimit the file name and file type with a period (fn.ft), or you can delimit them with a blank space. If you delimit them with a blank space, the file name and file type must be enclosed in matching quotation marks ('fn ft').
- On an MVS operating system, a file name is not case-sensitive. You can specify a sequential data set name, or a member of a PDS or PDSE that is from 1 to 44 alphanumeric characters. You can delimit the name with periods, using no more



than 8 characters between each period. Each set of 8 characters must start with a letter, A through Z.

In TSO, the normal TSO data set prefix rules apply. In general, the user ID is added only if the file name is not enclosed in quotation marks.

A single greater than sign (>) indicates that you want to write the output to a new file or write over an existing file.

For example, to write the output of a QUERY DOMAIN command to a new file, enter:

```
Query D0main acctg > dominfo.acc
```

Double greater than signs (>>) indicate that you want to append the output to the end of an existing file.

For example, to append the output of a QUERY DOMAIN command to the end of an existing file, enter:

```
Query D0main acctg >> dominfo.acc
```

Redirection characters must be preceded by and followed by at least one blank space. If a file with the name you specify does *not* exist, ADSM creates a new file with that name.

You can redirect output from any command. However, the most common use for redirecting output is to save output from query commands. The output from some query commands can be quite lengthy, and it can be easier for you to browse the contents of a file or, in some cases, print the contents of the file.

Normally, ADSM formats query output according to your screen or window width. If the data does not fit across the screen, then separate lines will be used for each output field. However, when you redirect the output of a query command, it is written to the output file with a line width of about 500 characters. This means that the output from query commands can format differently in redirected output. However, only the format of the output changes, the content is not modified.

## Using Macros to Issue Commands

A macro is a file that contains one or more ADSM administrative commands. You can only issue macros from the administrative client in the batch or interactive modes.

The name for a macro file must follow the naming conventions of the administrative client running on your operating system. For example, naming conventions for some operating systems are presented below.

- On AIX- and UNIX-based operating systems, a file name is case-sensitive, and can contain as many as 64 alphanumeric characters, with unlimited use of periods (.) for delimiting the file name.
- On DOS and Windows operating systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.

- On a Windows NT 3.5 operating system:
  - For high-performance file system (HPFS), a file name is not case-sensitive and can contain as many as 255 characters.
  - For file allocation table (FAT) file systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
  - For NT file system (NTFS), a file name can contain as many as 255 characters (including spaces and periods).
- On a Windows NT 4.0 operating system:
  - For file allocation table (FAT) file systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
  - For NT file system (NTFS), a file name can contain as many as 255 characters (including spaces and periods).
- On an OS/2 operating system:
  - For file allocation table (FAT) file systems, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an optional 3-character, alphanumeric extension.
  - For high-performance file system (HPFS), a file name is not case-sensitive and can contain as many as 255 characters.
- On a VM operating system, a file name is not case-sensitive. You can specify an 8-character, alphanumeric file name with an 8-character, alphanumeric file type. You can delimit the file name and file type with a period (fn.ft) or you can delimit them with a blank space. If you delimit them with a blank space, the file name and file type must be enclosed in matching quotation marks ('fn ft').
- On an MVS operating system, a file name is not case-sensitive. You can specify a data set name that is from 1 to 44 alphanumeric characters. You can delimit the name with periods, using no more than 8 characters between each period. Each set of 8 characters must start with a letter, A through Z.
 

In TSO, the normal TSO data set prefix rules apply. In general, the User ID is added only if the file name is not enclosed in quotation marks.

A macro invoked from the **adsm>** prompt is called a high-level macro. Any macros invoked from within the high-level macro are called *nested* macros. You can include the MACRO command within a macro file to invoke other macros up to ten levels deep.

Creating a macro file to enter commands can be especially helpful when you want to issue commands that are used repeatedly. For example, you can create a macro file that contains a command to define a storage pool. By using variables for the keyword values in the file, you can define a storage pool without having to type all the keyword parameters.

You can also enter commands in a macro file to process related commands in a specific order. For example, you could create a macro file that contains commands to define a new schedule and to associate client nodes to that schedule. Or, you could create a macro file that contains commands to define a new policy set, validate it, and make it the active policy set for a policy domain.

In a macro that contains several commands, you can use the COMMIT and ROLLBACK commands to control command processing within the macro. See “Controlling Command Processing in a Macro” on page 25 for information about using those commands.

### Writing Commands in a Macro File

When you write administrator commands in a macro, use the general rules stated in Chapter 2, “Entering Administrative Commands” on page 5. ADSM ignores any blank lines included in your macro. However, a completely blank line terminates a command that is continued (with continuation characters).

Refer to “Using Macros to Issue Commands” on page 21 for information about how to name a macro file.

Here is an example of a macro called REG.MAC that registers and grants authority to a new administrator:

```
REGister Admin pease mypasswd      -  
  CONTACT='David Pease, x1234'     -  
GRant AUTHority pease              -  
  CLasses=Policy,STorage           -  
  D0mains=domain1,domain2         -  
  STGpools=stgpool1,stgpool2
```

This example uses continuation characters in the macro file to issue the commands. For more information on continuation characters, see “Using Continuation Characters” on page 19.

After you create a macro file, you can update the information it contains and use it again, or you can copy the macro file, make changes to the copy, and then run the copy.

### Writing Comments in a Macro File

You can add comments to your macro file to document its purpose and functions.

To write a comment, write a slash and an asterisk (/\*) to indicate the beginning of the comment, write the comment, and then write an asterisk and a slash (\*/) to indicate the end of the comment. You can put a comment on a line by itself, or put it on a line that contains a command or part of a command.

For example, if you want to use a comment to identify the purpose of a macro, write the following:

```
/* AUTH.MAC-Register new nodes */
```

Or, if you want to write a comment to explain something about a command or part of a command, write:

```
DOMain=domain1          /*Assign node to DOMAIN1 */
```

Comments cannot be nested and cannot span lines. Every line of a comment must contain the comment delimiters.

## Using Substitution Variables in a Macro

You can use substitution variables in a macro if you want to supply values for commands when you run the macro. When you use substitution variables, you can use a macro again and again, whenever you need to perform the same task for different objects or with different parameter values.

A substitution variable consists of a percent sign (%), followed by a number that indicates the number of the substitution variable. When you run the file with the MACRO command, you must replace the numbered substitution variable with a string.

For example, if you want to create a macro named AUTH.MAC to register new nodes, write it as follows:

```
/* Register new nodes */
REGister Node %1 %2 -      /* userid password          */
  CONTACT=%3 -            /* 'name, phone number' */
  DOMain=%4               /* policy domain        */
```

Then, when you run the macro, you enter the values you want to pass to the server to process the command.

If your system uses the percent sign as a wildcard character, a pattern-matching expression in a macro where the percent sign is immediately followed by a numeric digit is interpreted by the administrative client as a substitution variable.

You cannot enclose a substitution variable in quotation marks. However, a value you supply as a substitution for the variable can be a quoted string.

For example, to register the node named DAVID with a password of DAVIDPW, with his name and phone number included as contact information, and assign him to the DOMAIN1 policy domain, enter:

```
MACRO auth.mac david davidpw "David Pease, x1234" domain1
```

## Running the Commands in a Macro

Table 5 identifies the command you use to run a macro.

Table 5. Running a Macro

Command	Description	Reference
MACRO	Runs a specified macro file.	Page 306

You can enter the MACRO command in batch or interactive mode.

If the macro does not contain substitution variables, (such as the REG.MAC macro described in the “Writing Commands in a Macro File” on page 23 ), to run the macro simply enter the MACRO command with the name of the macro file. For example:

```
MACRO reg.mac
```

If the macro contains substitution variables, (such as the AUTH.MAC macro described in “Using Substitution Variables in a Macro” on page 24), include the values that you want to supply after the name of the macro. Each value is delimited by a space.

For example:

```
MACRO auth.mac pease mypasswd "david pease, x1234" domain1
```

If you enter fewer values than there are substitution variables in the macro, ADSM replaces the remaining variables with null strings.

If you want to omit one or more values between values, enter a null string ("") for each omitted value. For example, if you omit the contact information in the previous example, you must enter:

```
MACRO auth.mac pease mypasswd "" domain1
```

### Controlling Command Processing in a Macro

When you issue a MACRO command, the server processes all commands in the macro file in order, including commands contained in any nested macros. The server commits all commands in a macro after successfully completing processing for the highest-level macro. If an error occurs in any command in the macro or in any nested macro, the server terminates processing and rolls back any changes caused by all previous commands.

If you specify the ITEMCOMMIT option when you enter the DSMADMC command, the server commits each command in a macro individually, after successfully completing processing for each command. If an error occurs, the server continues processing and only rolls back changes caused by the failed command.

You can control precisely when commands are committed with the COMMIT command. If an error occurs while processing the commands in a macro, the server terminates processing of the macro and rolls back any uncommitted changes (commands that have been processed since the last COMMIT). Make sure your administrative client session is *not* running with the ITEMCOMMIT option if you want to control command processing with the COMMIT command.

**Note:** Commands that start background processes cannot be rolled back. For a list of commands that can start background processes, see “Commands Processed in the Background” on page 7.

You can test a macro before implementing it by using the ROLLBACK command.

You can enter the commands (except the COMMIT command) you want to issue in the macro, and enter ROLLBACK as the last command. Then, you can run the macro to verify that all the commands process successfully. Any changes to the database caused by the commands are rolled back by the ROLLBACK command you have included at the end. Remember to remove the ROLLBACK command before you make the macro available for actual use. Also, make sure your administrative client session is *not* running with the ITEMCOMMIT option if you want to control command processing with the ROLLBACK command.

If you have a series of commands that process successfully via the command-line, but are unsuccessful when issued within a macro, there are probably dependencies between commands. It is possible that a command issued within a macro cannot be processed successfully until a previous command, that is issued within the same macro, is committed. Either of the following actions allow successful processing of these commands within a macro:

- Insert a COMMIT command before the command dependent on a previous command. For example, if COMMAND C is dependent upon COMMAND B you would insert a COMMIT command before COMMAND C. An example of this macro is:

```
COMMAND A  
COMMAND B  
COMMIT  
COMMAND C
```

- Start the administrative client session using the ITEMCOMMIT option. This causes each command within a macro to be committed before the next command is processed.

Table 6 identifies commands you can use to commit and roll back commands in a macro.

---

*Table 6. Committing or Rolling Back a Command in a Macro*

---

<b>Command</b>	<b>Description</b>	<b>Reference</b>
COMMIT	Makes changes to the database permanent.	Page 87
ROLLBACK	Discards any uncommitted changes to the database since the last COMMIT was executed.	Page 503

---





---

## Chapter 4. Using Commands Based on Privilege Class

In ADSM, there are five administrator privilege classes:

- System
- Policy
- Storage
- Operator
- Analyst

The authority granted to an administrator determines the administrative commands that the administrator can issue. After an administrator has been registered via the REGISTER ADMIN command, the administrator can issue a limited set of commands, including all query commands.

When you install ADSM, the server console is defined as a system administrator named SERVER\_CONSOLE and is granted system privilege. For information on how to issue commands from the SERVER\_CONSOLE administrative ID, refer to *ADSM Quick Start*.

The following sections describe each type of administrator privilege and the commands that can be issued by an administrator who has been granted the corresponding authority.

---

### Commands Requiring System Privilege

An administrator with system privilege has the highest level of authority in ADSM. With system privilege an administrator can issue any administrative command and has authority to manage all policy domains and all storage pools.

Table 7 lists the commands that only administrators with system privilege can issue.

---

*Table 7 (Page 1 of 4). System Privilege Commands*

Command	Description	Reference
AUDIT LICENSES	Checks for compliance with defined licenses.	Page 54
CANCEL PROCESS	Cancels a background server process.	Page 74
COPY DOMAIN	Creates a copy of an existing policy domain.	Page 89
COPY SCHEDULE (See note.)	Creates a copy of an existing schedule.	Page 96
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144

---

Table 7 (Page 2 of 4). System Privilege Commands

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE SCHEDULE (See note.)	Defines a schedule for a client operation or an administrative command.	Page 164
DEFINE STGPOOL	Defines a storage pool as a named collection of data storage media.	Page 180
DELETE DOMAIN	Deletes a policy domain along with any policy objects in the policy domain.	Page 208
DELETE SCHEDULE (See note.)	Deletes a schedule from the database.	Page 229
DELETE STGPOOL	Deletes a storage pool from data storage.	Page 232
EXPIRE INVENTORY	Manually starts inventory expiration processing.	Page 248
EXPORT ADMIN	Copies administrative information to external media.	Page 250
EXPORT NODE	Copies client node information to external media.	Page 254
EXPORT POLICY	Copies policy information to external media.	Page 260
EXPORT SERVER	Copies all or part of the server to external media.	Page 264
GRANT AUTHORITY	Assigns one or more privilege classes to an administrator.	Page 273
IMPORT ADMIN	Restores administrative information from external media.	Page 283
IMPORT NODE	Restores client node information from external media.	Page 287
IMPORT POLICY	Restores policy information from external media.	Page 293
IMPORT SERVER	Restores all or part of the server from external media.	Page 297
LOCK ADMIN	Prevents an administrator from accessing ADSM.	Page 303

Table 7 (Page 3 of 4). System Privilege Commands

Command	Description	Reference
REGISTER ADMIN	Defines a new administrator without granting administrative authority.	Page 461
REGISTER LICENSE	Registers a new license with the ADSM server.	Page 463
REMOVE ADMIN	Removes an administrator from the list of registered administrators.	Page 470
RENAME ADMIN	Changes an ADSM administrator's name.	Page 474
REVOKE AUTHORITY	Revokes one or more privilege classes or restricts access to policy domains and storage pools.	Page 498
SET ACCOUNTING	Specifies whether accounting records are created at the end of each client session.	Page 505
SET ACTLOGRETENTION	Specifies the number of days to retain log records in the activity log.	Page 507
SET AUTHENTICATION	Specifies whether clients are prompted for a password at the beginning of a session.	Page 509
SET EVENTRETENTION	Specifies the number of days to retain event records.	Page 510
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.	Page 512
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514
SET MAXCMDRETRIES	Specifies the maximum number of retries after a failed attempt to execute a scheduled command.	Page 517

Table 7 (Page 4 of 4). System Privilege Commands

Command	Description	Reference
SET MAXSCHEDESESSIONS	Specifies the maximum number of client/server sessions available for processing scheduled work.	Page 519
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.	Page 521
SET QUERYSCHEDPERIOD	Specifies the frequency for clients to obtain scheduled work, in client-polling mode.	Page 524
SET RANDOMIZE	Specifies the randomization of start times within a window for schedules in client-polling mode.	Page 526
SET REGISTRATION	Specifies whether users can register themselves or must be registered by an administrator.	Page 528
SET RETRYPERIOD	Specifies the time between retry attempts by the client scheduler.	Page 530
SET SCHEDMODES	Specifies the central scheduling mode for the server.	Page 532
SET SERVERNAME	Specifies the name by which the server is identified.	Page 534
UNLOCK ADMIN	Enables a locked administrator to access ADSM.	Page 535
UPDATE ADMIN	Changes the password or contact information associated with any administrator.	Page 537
UPDATE SCHEDULE (See note.)	Changes the attributes of a schedule.	Page 590

**Note:** Indicates that this command is restricted by the authority granted to an administrator. Storage privilege is required only for administrative command schedules. Policy privilege is required for client operation schedules.

---

## Commands Requiring Policy Privilege

An administrator with policy privilege can issue commands that relate to policy management objects such as policy domains, policy sets, management classes, copy groups, and schedules. The policy domains that an administrator can manage depend on the authority granted to them by an administrator with system privilege.

As an administrator with policy privilege, you can have unrestricted or restricted policy privilege.

Unrestricted policy privilege permits you to issue all of the administrator commands that require policy privilege. You can issue commands that affect all existing policy domains as well as any policy domains that are defined in the future. An unrestricted policy administrator cannot define, delete, or copy policy domains.

Restricted policy privilege permits you to issue administrator commands that affect one or more policy domains for which you have been explicitly granted authority. For example, the DELETE MGMTCLASS command requires you to have policy privilege for the policy domain to which the management class belongs.

Table 8 lists the commands that an administrator with policy privilege can issue.

---

*Table 8 (Page 1 of 3). Policy Privilege Commands*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
ASSIGN DEFMGMTCLASS	Assigns an existing management class as the default for a specified policy set.	Page 48
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
COPY POLICYSET	Creates a copy of an existing policy set.	Page 94
COPY SCHEDULE (See note 2.)	Creates a copy of an existing schedule.	Page 96
DEFINE ASSOCIATION	Associates one or more clients with a schedule.	Page 100
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103

---

Table 8 (Page 2 of 3). Policy Privilege Commands

Command	Description	Reference
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DEFINE SCHEDULE (See note 2.)	Defines a schedule for a client operation or an administrative command.	Page 164
DELETE ASSOCIATION	Deletes the association between one or more clients and a schedule.	Page 197
DELETE COPYGROUP	Deletes a backup or archive copy group from a policy domain and policy set.	Page 199
DELETE EVENT (See note 1.)	Deletes event records prior to a specified date and time.	Page 212
DELETE FILESPACE	Deletes data associated with one or more client's filespaces.	Page 216
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
DELETE SCHEDULE (See note 2.)	Deletes a schedule from the database.	Page 229
LOCK NODE	Prevents a client from accessing the server.	Page 304
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466
REMOVE NODE	Removes a client from the list of registered nodes for a specific policy domain.	Page 472
RENAME FILESPACE	Renames a client filesystem on the server.	Page 476

Table 8 (Page 3 of 3). Policy Privilege Commands

Command	Description	Reference
RENAME NODE	Changes the name for an existing user that belongs to a specific policy domain.	Page 479
UNLOCK NODE	Enables a locked user in a specific policy domain to access the server.	Page 536
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539
UPDATE DOMAIN	Changes the attributes of a policy domain.	Page 573
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581
UPDATE NODE	Changes the attributes associated with a client node.	Page 585
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
UPDATE SCHEDULE (See note 2.)	Changes the attributes of a schedule.	Page 590
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

**Notes:**

1. Indicates that this command cannot be restricted by policy domain. An administrator with unrestricted policy privilege or restricted policy privilege for a specified policy domain can issue this command.
2. Indicates that this command is restricted by the authority granted to an administrator. Storage privilege is required only for administrative command schedules. Policy privilege is required for client operation schedules.

## Commands Requiring Storage Privilege

An administrator with storage privilege can issue commands that allocate and control storage resources for the server. The commands that an administrator can issue depend on the authority granted to them by an administrator with system privilege. As an administrator with storage privilege, you can have unrestricted or restricted storage privilege.

Unrestricted storage privilege permits you to issue all of the administrator commands that require storage privilege. You can issue commands that affect all existing storage

pools as well as any storage pools that are defined in the future. You can also issue commands that affect the database and the recovery log. An unrestricted storage administrator cannot define or delete storage pools.

Restricted storage privilege permits you to issue administrator commands that only affect a storage pool for which you have been granted authority. For example, the DELETE VOLUME command only affects a storage pool volume that is defined to a specific storage pool.

Table 9 lists the commands an administrator with storage privilege can issue.

*Table 9 (Page 1 of 3). Storage Privilege Commands*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
AUDIT VOLUME (See note.)	Compares database and storage pool information, and optionally, resolves any inconsistencies.	Page 56
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
BACKUP DEVCONFIG	Backs up ADSM device information to a file.	Page 66
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
BACKUP VOLHISTORY	Records sequential volume history information in external files.	Page 72
CHECKIN LIBVOLUME	Checks a storage volume into an automated library.	Page 80
CHECKOUT LIBVOLUME	Checks a storage volume out of an automated library.	Page 84
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118



Table 9 (Page 2 of 3). Storage Privilege Commands

Command	Description	Reference
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
DEFINE VOLUME (See note.)	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE DBBACKUPTRIGGER	Deletes the database backup trigger.	Page 202
DELETE DBVOLUME	Deletes a volume from the database.	Page 204
DELETE DEVCLASS	Deletes a device class name.	Page 206
DELETE DRIVE	Deletes a drive from a library.	Page 210
DELETE LIBRARY	Deletes a library.	Page 220
DELETE LOGVOLUME	Deletes a volume from the recovery log.	Page 222
DELETE VOLHISTORY	Removes sequential volume history information from the volume history file.	Page 234
DELETE VOLUME (See note.)	Deletes a volume from a storage pool.	Page 240
EXTEND DB	Extends usable database storage within the volumes previously allocated for its use.	Page 269
EXTEND LOG	Extends usable log storage within the volumes previously allocated for its use.	Page 271

Table 9 (Page 3 of 3). Storage Privilege Commands

Command	Description	Reference
MOVE DATA (See note.)	Moves data from a specified storage pool volume to another storage pool volume.	Page 309
REDUCE DB	Attempts to free up a database volume by reducing usable storage within the volumes defined.	Page 457
REDUCE LOG	Attempts to free up a log volume by reducing usable storage within the volumes defined.	Page 459
RESTORE STGPOOL	Restores files to a primary storage pool from copy storage pools.	Page 488
RESTORE VOLUME	Restores files stored on specified volumes in a primary storage pool from copy storage pools.	Page 493
UPDATE DBBACKUPTRIGGER	Changes one or more attributes of the database backup trigger.	Page 547
UPDATE DEVCLASS	Changes the attributes of a device class.	Page 550
UPDATE DRIVE	Changes the attributes of a drive.	Page 575
UPDATE LIBRARY	Changes the attributes of a library.	Page 577
UPDATE STGPOOL (See note.)	Changes the attributes of a storage pool.	Page 605
UPDATE VOLHISTORY	Adds or changes location information for a volume in the volume history file.	Page 617

**Note:** Indicates that this command can be restricted by storage pool. An administrator with unrestricted storage privilege or restricted storage privilege for a specified storage pool can issue this command.

## Commands Requiring Operator Privilege

An administrator with operator privilege can issue commands that control the immediate operation of the server and the availability of storage media.

Table 10 on page 39 lists the commands an administrator with operator privilege can issue.

*Table 10. Operator Privilege Commands*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL REQUEST	Cancels one or more pending volume mount requests.	Page 76
CANCEL SESSION	Cancels one or all active sessions with the server.	Page 78
DISABLE	Prevents new clients from accessing ADSM but permits existing sessions to continue.	Page 243
DISMOUNT VOLUME	Dismounts a sequential, removable volume by the volume name.	Page 245
ENABLE	Resumes server activity following the DISABLE command.	Page 246
HALT	Shuts down the server immediately. (May be preceded by the DISABLE command.)	Page 277
REPLY	Allows a request to continue processing.	Page 481
UPDATE VOLUME	Updates the attributes of one or more storage pool volumes.	Page 619
VARY	Specifies whether a disk volume is available to the server for use.	Page 629

## Commands Requiring Analyst Privilege

An administrator with analyst privilege can issue commands that reset the counters that track server statistics.

Table 11 lists the commands an administrator with analyst privilege can issue.

*Table 11 (Page 1 of 2). Analyst Privilege Commands*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
RESET BUFPOOL	Resets the buffer pool statistics for the database.	Page 482

---

*Table 11 (Page 2 of 2). Analyst Privilege Commands*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
RESET DBMAXUTILIZATION	Resets the maximum utilization statistic for the database.	Page 484
RESET LOGCONSUMPTION	Resets the statistic on recovery log space used.	Page 485
RESET LOGMAXUTILIZATION	Resets the maximum utilization statistic for the recovery log.	Page 487

---

## **Commands Any Administrator Can Issue**

Table 12 on page 41 lists the commands any registered administrator can issue even if that administrator has not been granted any specific administrator privileges.

Table 12 (Page 1 of 3). Commands Issued by All Administrators

Command	Description	Reference
COMMIT	Makes changes to the database permanent.	Page 87
HELP	Displays help for administrative commands.	Page 280
MACRO	Runs a specified macro file.	Page 306
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
QUERY ASSOCIATION	Displays the clients associated with one or more schedules.	Page 322
QUERY AUDITOCUPANCY	Displays the data storage utilization for a client node.	Page 325
QUERY CONTENT	Displays information about one or more files currently residing in a storage pool volume.	Page 328
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY DB	Displays allocation information about the database.	Page 339
QUERY DBBACKUPTRIGGER	Displays information about the database backup trigger.	Page 345
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347
QUERY DEVCLASS	Displays information about one or more device classes.	Page 350
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
QUERY DRIVE	Displays information about one or more drives.	Page 359

Table 12 (Page 2 of 3). Commands Issued by All Administrators

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY EVENT	Displays information about scheduled and completed events for selected clients.	Page 362
QUERY FILESPACE	Displays information about data in one or more filespaces that belong to a client.	Page 377
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
QUERY LIBVOLUME	Displays information about a library volume.	Page 384
QUERY LICENSE	Displays information about licenses and audits.	Page 386
QUERY LOG	Displays allocation information about the recovery log.	Page 389
QUERY LOGVOLUME	Displays information about one or more log volumes.	Page 394
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
QUERY MOUNT	Displays information about mounted sequential access media.	Page 401
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403
QUERY OCCUPANCY	Displays file space information by storage pool.	Page 408
QUERY OPTION	Displays information about one or more options from the server options file.	Page 412
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415

Table 12 (Page 3 of 3). Commands Issued by All Administrators

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY REQUEST	Displays information about all pending mount requests.	Page 421
QUERY SCHEDULE	Displays information about one or more schedules.	Page 422
QUERY SESSION	Displays information about all active administrator and client sessions with ADSM.	Page 427
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
QUERY STGPOOL	Displays information about one or more storage pools.	Page 436
QUERY VOLHISTORY	Displays sequential volume history information that has been collected by the server.	Page 443
QUERY VOLUME	Displays information about one or more storage pool volumes.	Page 450
QUIT	Ends an interactive administrative client session.	Page 456
ROLLBACK	Discards any uncommitted changes to the database since the last COMMIT was executed.	Page 503





---

## Chapter 5. Using Administrator Commands

This chapter describes the ADSM administrator commands.

Information for each command includes:

- A description of the tasks a command performs
- The administrator privilege class required to use the command
- A syntax diagram that identifies the required and optional parameters for the command
- Descriptions of each parameter of the command
- Usage notes (when appropriate)
- Examples of using the command
- A list of related commands

## ACTIVATE POLICYSET

---

### ACTIVATE POLICYSET (Activate a New Policy Set)

Use the ACTIVATE POLICYSET command to specify an existing policy set as the ACTIVE policy set for a policy domain.

The ACTIVATE POLICYSET command will fail if the following conditions exist:

- A copy group within the policy set specifies a copy storage pool as a destination.
- A management class specifies a copy pool as the destination for space-managed files.
- A default management class is not defined.

When a policy set is activated, the contents of the policy set are copied to a policy set that has the reserved name ACTIVE. Once activated, there is no real relationship between the policy set that has been activated (copied to ACTIVE) and the contents of the ACTIVE policy set. The original policy set can still be modified, but the copied definitions in the ACTIVE policy set can only be modified by activating another policy set.

Because of this copying process, you should not assume that the definitions in the policy set that was last activated are the same as those in the ACTIVE policy set; the original policy set may have been modified since activation.

You can define multiple policy sets for a policy domain, but only one policy set can be active at a time. The current active policy set is replaced by the one you specify when you issue this command.

Prior to activating a policy set, issue the VALIDATE POLICYSET command. This command verifies that a policy set is complete and valid prior to activation.

You can use the QUERY DOMAIN command to find out which policy set was last activated.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

### Syntax

```
▶▶—ACTivate POLicyset—domainname—setname————▶▶
```

## ACTIVATE POLICYSET

### Parameters

*domainname*

Specifies the name of the policy domain for the policy set you want to activate. This parameter is required.

*setname*

Specifies the name of policy set you want to activate. This parameter is required.

### Examples

#### Task

Activate the VACATION policy set in the EMPLOYEE\_RECORDS policy domain.

**Command:** `activate policyset employee_records vacation`

#### Output:

```
ANR1514I Policy set VACATION activated in policy domain  
EMPLOYEE_RECORDS.
```

### Related Commands

Table 13 provides references to related commands.

*Table 13. Commands Related to ACTIVATE POLICYSET*

Command	Description	Reference
COPY POLICYSET	Creates a copy of an existing policy set.	Page 94
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## ASSIGN DEFMGMTCLASS

---

### ASSIGN DEFMGMTCLASS (Assign a Default Management Class)

Use the ASSIGN DEFMGMTCLASS command to specify an existing management class as the default management class for a particular policy set. You must assign a default management class for a policy set before you can activate that policy set. It is recommended that the default management class contain both an archive copy group and a backup copy group.

ADSM uses the default management class when:

- A user does not specify a management class in their include-exclude list. The server then uses the default management class to manage files when they are backed up or archived.
- The management class associated with a backup copy or archived file is deleted. If the default management class does not contain a backup or archive copy group, the retention grace period for the policy domain is used.
- The backup copy group is deleted from a management class; backed up files that are bound to that management class are then managed with the default management class (assuming that the default management class contains a backup copy group). If the default management class does not contain a backup or archive copy group, the retention grace period for the policy domain is used.
- The archive copy group is deleted from a management class; archived files that are bound to that management class are then managed with the default management class (assuming that the default management class contains an archive copy group). If the default management class does not contain an archive copy group, the archive retention grace period for the policy domain is used.
- A client node is reassigned to a different policy domain, and the active policy set does not contain management classes with the same name as that to which the node's backed up files were bound. The files are bound to the default management class if the management class contains a backup copy group or an archive copy group. If the default management class does not contain these copy groups, the retention grace period is used during expiration.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

### Syntax

```
▶▶—ASsIgn DEFMGmtclass—domainname—setname—classname—▶▶
```

## ASSIGN DEFMGMTCLASS

### Parameters

*domainname*

Specifies the name of the policy domain to which the management class belongs. This parameter is required.

*setname*

Specifies the name of the policy set associated with the management class to be specified as the default management class. This parameter is required. You cannot assign a default management class to the ACTIVE policy set.

*classname*

Specifies the name of the management class to be specified as the default management class. This parameter is required.

### Examples

#### Task

Assign DEFAULT1 as the default management class for policy set SUMMER in the PROG1 policy domain.

**Command:** `assign defmgmtclass prog1 summer default1`

#### Output:

```
ANR1538I Default management class set to DEFAULT1 for policy domain PROG1, set SUMMER.
```

### Related Commands

Table 14 provides references to related commands.

Table 14 (Page 1 of 2). Commands Related to ASSIGN DEFMGMTCLASS

Command	Description	Reference
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157

## ASSIGN DEFMGMTCLASS

Table 14 (Page 2 of 2). Commands Related to ASSIGN DEFMGMTCLASS

Command	Description	Reference
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## AUDIT LIBRARY

### AUDIT LIBRARY (Verify an Automated Library's Inventory)

Use the AUDIT LIBRARY command to ensure that an automated library is in a consistent state with respect to the server's internal library volume inventory. If inconsistencies are detected, the server updates its inventory to reflect the current state of the library.

**Important:** Audit library processing waits until all volumes have been demounted from drives within the specified library. If one or more volumes are mounted, but are in the IDLE state, you can force the volumes to be demounted by issuing the DISMOUNT VOLUME command. Otherwise, the audit library operation remains in a wait state until the idle volumes have been demounted (the idle volumes are demounted once the MOUNTRETENTION period expires).

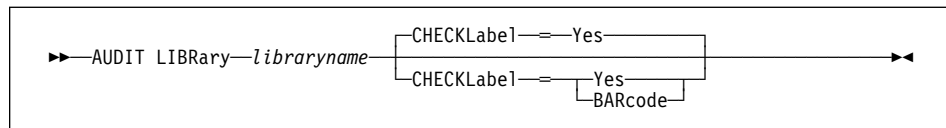
If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command.

This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### *libraryname*

Specifies the name of the library to be audited. This parameter is required.

#### **CHECKLabel=***chkblvalue*

Specifies how the storage volume label is checked during audit library processing. This parameter is optional and is only applicable to SCSI libraries. The default is YES. Possible values are:

#### **Yes**

Specifies to use the storage volume label on the tape.

#### **BARcode**

Specifies to use the barcode storage volume label. You can decrease the audit processing time by using the barcode.

## AUDIT LIBRARY

### Usage Notes

This command prevents any other activity with the library until the background process it starts is done. For example, no restore or retrieve requests that involve this library are processed while this command runs.

### Examples

#### Task

Ensure that the EZLIFE automated library is in a consistent state with respect to the ADSM server's library volume inventory.

**Command:** `AUDIT LIBRARY ezlife`



## AUDIT LIBRARY

### Output:

```
ANR8457I AUDIT LIBRARY: Operation for library EZLIFE
started as process 5.
ANR8461I AUDIT LIBRARY process for library EZLIFE
completed successfully.
```

### Related Commands

Table 15 provides references to related commands.

*Table 15. Commands Related to AUDIT LIBRARY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE LIBRARY	Deletes a library.	Page 220
DISMOUNT VOLUME	Dismounts a sequential, removable volume by the volume name.	Page 245
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
QUERY LIBVOLUME	Displays information about a library volume.	Page 384
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
UPDATE LIBRARY	Changes the attributes of a library.	Page 577

## AUDIT LICENSES

---

### AUDIT LICENSES (Audit Data Storage Usage)

Use the AUDIT LICENSES command to start a process that both audits the data storage used by each client node and licensing features in use on the server. This process then compares the storage utilization and other licensing factors to the license terms that have been defined to the server to determine if the current server configuration is in compliance with the license terms.

**Note:** If you halt the server and then restart it, a license audit is run automatically as specified by the SET LICENSEAUDITPERIOD command to check for license conformance.

If the audit process identifies that the server configuration is not in compliance with the terms of the license, one of the following occurs:

- Warning messages are generated indicating that the server is not in compliance with the license terms. However, operations continue normally.
- Operations fail because the server is not licensed for specific features or device support modules.

After the license audit is complete, you can view the information that has been gathered by using the QUERY LICENSE command.

As part of a license audit operation, the server calculates, by node, the amount of backup, archive, and space management in use. For servers managing large amounts of data, this calculation can take a great deal of CPU time and can stall other server activity. You can use the NOAUDITSTORAGE option to specify that storage is not to be calculated as part of a license audit.

The AUDIT LICENSES command also may be used after new licenses have been registered with the REGISTER LICENSE command so that the new license terms are known to the server before the next automatic license audit is run.

This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
▶▶—AUDit LICenses—▶▶
```

### Parameters

None.

## AUDIT LICENSES

### Examples

#### Task

Issue the AUDIT LICENSES command.

**Command:** AUDit LICenses

#### Output:

```
ANR2817I AUDIT LICENSES: License audit started as
process 2.

ANR2825I License audit process 2 completed successfully -
10 nodes audited.
ANR2811I Audit License completed - Server is in compliance
with license terms.
```

### Related Commands

Table 16 provides references to related commands.

*Table 16. Commands Related to AUDIT LICENSES*

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
QUERY AUDITOCUPANCY	Displays the data storage utilization for a client node.	Page 325
QUERY LICENSE	Displays information about licenses and audits.	Page 386
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
REGISTER LICENSE	Registers a new license with the ADSM server.	Page 463
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.	Page 512

## AUDIT VOLUME

---

### AUDIT VOLUME (Verify Database Information for a Storage Pool Volume)

Use the AUDIT VOLUME command to check for any inconsistencies between database information and a storage pool volume. During the audit process, the server sends processing information to the activity log and the server console.

While an audit process is active, clients cannot restore data from the specified volume or store new data to that volume.

This command can be used to audit a volume assigned either to a primary storage pool or to a copy storage pool. If a file on a volume is detected with integrity errors, handling of the file will depend on the type of storage pool to which the volume belongs, whether the FIX option specified on this command, and whether the file is also stored on a volume assigned to other pools.

You cannot audit a volume if the volume specified on this command is being deleted from a primary or copy storage pool.

If the AUDIT VOLUME command does not detect data-integrity errors for a file that had previously been marked as *damaged*, the state of the file is reset so that the file can be used. This provides a means for resetting the state of *damaged* files if it is determined that the errors were caused by correctable hardware problems such as a dirty tape head.

To display damaged files on a specific volume, use the QUERY CONTENT command.

Do not use the AUDIT VOLUME command with FIX=YES if a restore process (RESTORE STGPOOL or RESTORE VOLUME) is running. The AUDIT VOLUME command could cause the restore to be incomplete.

To display information about the contents of a storage pool volume, use the QUERY CONTENT command.

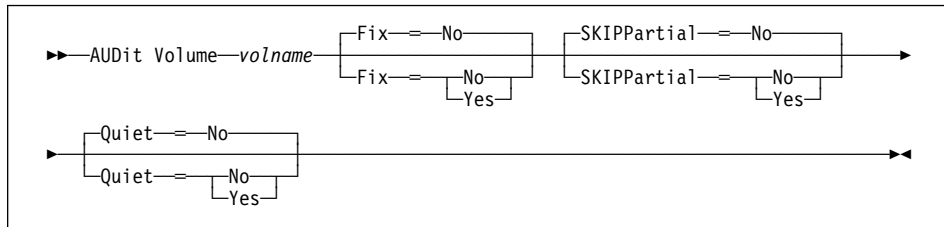
This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool to which the volume is defined.

## AUDIT VOLUME

### Syntax



### Parameters

#### *volname*

Specifies the name of the storage pool volume you want to audit. This parameter is required.

#### **Fix=fixvalue**

Specifies how ADSM resolves any logical inconsistencies between the database inventory and the specified storage pool volume. This parameter is optional. The default value is NO.

The actions ADSM performs based on this parameter also depend on whether the volume is assigned to a primary or a copy storage pool.

#### **Primary Storage Pool**

For a volume in a primary storage pool, the value for the FIX parameter has the following effects.

#### **Fix=No**

ADSM reports, but does not delete, any database records that refer to files found with logical inconsistencies.

If the AUDIT VOLUME command detects a data-integrity error in a file:

- ADSM marks the file as *damaged* in the database. If a backup copy is stored in a copy storage pool, the file can be restored using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the file is a cached copy, references to the file on this volume can be deleted by issuing the AUDIT VOLUME command and specifying FIX=YES.

If the AUDIT VOLUME command does not detect a data-integrity error in a file that had previously been marked as damaged, the state of the file is reset so that the file can be used. This provides a means for resetting the state of damaged files if it is determined that the errors were caused by correctable hardware problems such as a dirty tape head.

#### **Fix=Yes**

ADSM fixes any inconsistencies as they are detected.

If the AUDIT VOLUME command detects a data-integrity error in a file:

## AUDIT VOLUME

- If a backup copy is not stored in a copy storage pool, ADSM deletes all database records that refer to the file.
- If a backup copy is stored in a copy storage pool, ADSM marks the file as damaged in the database. The file can then be restored using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the file is a cached copy, ADSM deletes the database records that refer to the cached file. The primary file is stored on another volume.

If the AUDIT VOLUME command does not detect a data-integrity error in a file that had previously been marked as damaged, ADSM resets the state of the file so that it can be used. This provides a means for resetting the state of damaged files if it is determined that the errors were caused by correctable hardware problems such as a dirty tape head.

### Copy Storage Pool

For a volume in a copy storage pool, the value for the FIX parameter has the following effect if ADSM detects a file with integrity errors:

#### Fix=No

The error is reported and the file copy is marked as *damaged* in the database.

#### Fix=Yes

References to the file on the audited volume are deleted. ADSM deletes any database records that point to a file on the audited volume that does not exist.

#### SKIPPARTIAL=*skipvalue*

Specifies whether ADSM ignores files that span multiple storage pool volumes (*skipped files*), when performing the audit for the specified storage pool volume. This parameter is optional. The default value is NO. When performing an audit operation on a sequential access media volume, this parameter prevents additional sequential access media mounts that may be necessary to audit any *skipped files*. Possible values are:

#### No

Specifies that ADSM audits files that span multiple volumes when performing an audit operation on a specified volume. For sequential access volumes, multiple volumes can be mounted.

#### Yes

Specifies that ADSM only audits complete files that are stored on the volume to be audited. For sequential access volumes, only the specified volume is mounted. The status of any *skipped files* is unknown.

#### Quiet=*quietvalue*

Specifies whether ADSM sends detailed informational messages to the activity log and the server console about irretrievable files on the volume. This parameter is optional. The default value is NO. Possible values are:

## AUDIT VOLUME

### No

Specifies that ADSM sends detailed informational messages and a summary report to the activity log and the server console. Each message contains the node, file space, and client name for the file.

### Yes

Specifies that ADSM only sends a summary report to the activity log and the server console.

## Usage Notes

Unless you specify SKIPPARTIAL=YES, ADSM attempts to process each file stored on the volume, including files that span into and out of other volumes. In order to audit files that span multiple volumes, the following conditions must be true:

- For sequential access volumes, the additional sequential access volumes must have an access mode of *read/write* or *read-only*.
- For random access volumes, the additional volumes must be online.

## Examples

### Task

Verify that the database information for storage pool volume PROG2 is consistent with the data stored on the volume. ADSM fixes any inconsistencies that are found.

**Command:** AUDit Volume prog2 Fix=Yes

### Output:

```
ANR1199I Removable volume PROG2 is required for
Audit process.
ANR8324I 8MM volume PROG2 is expected to be
mounted (R/W).
ANR2313I Audit Volume (Inspect Only) process started
for volume PROG2 (process ID 206).
ANR2315I Audit Volume process ended for volume PROG2;
5 files inspected, 0 damaged files found and marked
as damaged.
```

## Related Commands

Table 17 provides references to related commands.

Table 17 (Page 1 of 2). Commands Related to AUDIT VOLUME

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74

## AUDIT VOLUME

*Table 17 (Page 2 of 2). Commands Related to AUDIT VOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY CONTENT	Displays information about one or more files currently residing in a storage pool volume.	Page 328
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY VOLUME	Displays information about one or more storage pool volumes.	Page 450



---

### BACKUP DB (Back Up the Database)

Use the BACKUP DB command to back up an ADSM database to sequential access storage volumes. You can use this command to run one of the following types of backup:

#### **Full backup**

Copies the entire ADSM database

#### **Incremental backup**

Copies only those database pages that have been added or changed since the last time the database was backed up

If the database is damaged or lost, you can restore it by using the DSMSERV RESTORE DB command to perform one of the following types of recovery:

#### **Roll-forward recovery**

Restores the database or a single database volume to its most current state by first loading the most recent database backup series (the last full backup and all incremental backups that apply to that full backup), and then using recovery log records to apply any changes made to the database since the last backup was created. The recovery log mode must have been set to ROLLFORWARD mode continuously from the time that the last backup series was created, until the time that the database was lost or damaged. A volume history file must be available. A device configuration file must be available or must be manually created.

#### **Point-in-time recovery**

Restores the database to the point in time when a specific backup was created by using information stored in the volume history file to locate the appropriate volumes or by using the volumes you specify.

A volume history file and a device configuration file are required for roll-forward recovery. The server options file that is created during installation includes VOLUMEHISTORY and DEVCONFIG options that specify default files named volhist.out and devcnfg.out, respectively. You can also use the BACKUP VOLHISTORY and BACKUP DEVCONFIG commands.

To help you determine how much storage space a full or incremental backup will require, use the QUERY DB command. That command displays the number of changed megabytes in the database.

**Note:** If the recovery log runs out of space during a database backup and the log mode is set to ROLLFORWARD, all ADSM transactions in progress are ended and rolled back. No new transactions may begin until the backup is complete and recovery log records are deleted. After the backup is successfully completed, ADSM deletes all recovery log records except those required to restore the database to its most current state when used with the backup of the database that was just created.

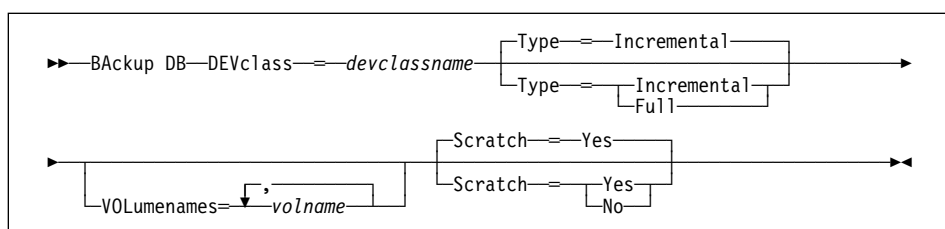
This command generates a background process that can be canceled with the CANCEL PROCESS command. If a BACKUP DB background process is canceled, some of the database may have already been backed up before the cancellation. To display information on background processes, use the QUERY PROCESS command.

## BACKUP DB

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### **DEVclass**=*devclassname*

Specifies the name of the sequential access device class to use for the backup.

The server options file that is created during installation includes a DEVCONFIG option that specifies an external file (devcnfg.out). ADSM stores a backup copy of device class definitions in this file. If your ADSM database is damaged or lost and must be restored, the definitions created by using the DEFINE DEVCLASS command will not be available and will need to be recreated manually. This parameter is required.

If the mount limit for the device class specified has already been reached (all drives are busy) when the backup runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the backup.

#### **Type**=*typevalue*

Specifies the type of backup to run. This parameter is optional. The default value is INCREMENTAL. Possible values are:

##### **Incremental**

Specifies that you want to run an incremental backup of the ADSM database. The maximum number of incremental backups you can run between full backups is 32.

Each incremental backup, whether run automatically by ADSM based on the settings you specify with the DEFINE DBBACKUPTRIGGER command or run by issuing the BACKUP DB command, is added to the count of incremental backups run.

You can run an incremental backup whenever a full backup is not required.

##### **Full**

Specifies that you want to run a full backup of the ADSM database. A full backup is required when any of the following conditions are true:

- The database has never been backed up.

## BACKUP DB

- The database has been restored using a salvage dump created with the DSMSERV DUMPDB command and has not been backed up since it was restored.
- The maximum number of incremental backups allowed between full backups has been reached.
- The log mode was changed from NORMAL to ROLLFORWARD.
- The database was extended or reduced.

Each full backup, whether run automatically by ADSM based on the settings you specify with the DEFINE DBBACKUPTRIGGER command or run by issuing the BACKUP DB command, resets the count for incremental backups to 0.

**Note:** Database backups (full or incremental) are not run automatically unless both the database backup trigger is defined *and* the recovery log mode is set to ROLLFORWARD.

### **VOLUMENAMES=volname**

Specifies the volumes to use for the backup. You can specify more than one volume by separating each volume name with a comma, with no intervening spaces.

This parameter is optional. However, if you specify SCRATCH=NO, you must specify a list of volumes.

### **SCRATCH=scratchvalue**

Specifies whether scratch volumes can be used for the backup. This parameter is optional. The default value is YES. Possible values are:

#### **Yes**

Specifies that scratch volumes can be used.

If you include a list of volumes by using the VOLUMENAMES parameter and specify SCRATCH=YES, ADSM uses scratch volumes only if not enough space is available to store the backup data on the specified volumes.

If you do not include a list of volumes by using the VOLUMENAMES parameter, you must either specify SCRATCH=YES or omit the SCRATCH parameter (to use the default).

#### **No**

Specifies that scratch volumes cannot be used.

If you specify volumes by using the VOLUMENAMES parameter and specify SCRATCH=NO, the backup will fail if not enough space is available to store the backup data on the specified volumes.

## BACKUP DB

### Examples

#### Task

Run an incremental backup of the database, using a scratch volume. Assume a device class of FILE for the backup.

**Command:** Backup DB DEVclass=file Type=Incremental

#### Output:

```
ANR2281I Incremental database backup started as process 4.
```

### Related Commands

Table 18 provides references to related commands.

*Table 18 (Page 1 of 2). Commands Related to BACKUP DB*

Command	Description	Reference
BACKUP DEVCONFIG	Backs up ADSM device information to a file.	Page 66
BACKUP VOLHISTORY	Records sequential volume history information in external files.	Page 72
CANCEL PROCESS	Cancel a background server process.	Page 74
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
DELETE DBBACKUPTRIGGER	Deletes the database backup trigger.	Page 202
DSMSERV RESTORE DB	Restores an ADSM database or database volume.	Page 653
QUERY DB	Displays allocation information about the database.	Page 339
QUERY DBBACKUPTRIGGER	Displays information about the database backup trigger.	Page 345
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514

## BACKUP DB

*Table 18 (Page 2 of 2). Commands Related to BACKUP DB*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
UPDATE DBBACKUPTRIGGER	Changes one or more attributes of the database backup trigger.	Page 547

## BACKUP DEVCONFIG

---

### BACKUP DEVCONFIG (Create Backup Copies of Device Configuration Information)

Use the BACKUP DEVCONFIG command to specify one or more files in which to store a backup copy of the following information:

- Device class definitions created using the DEFINE DEVCLASS command
- Library definitions created using the DEFINE LIBRARY command
- Drive definitions created using the DEFINE DRIVE command

You can run this command at any time to ensure that device configuration information is available.

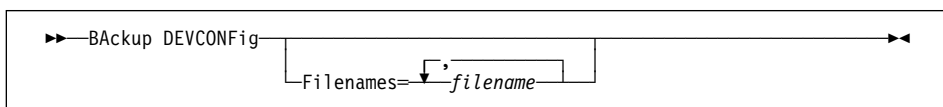
If your ADSM database is damaged or lost and must be restored using a database backup series or an offline salvage dump, device configuration information must be available before ADSM can restore the database.

The server options file that is created during installation includes a DEVCONFIG options that specifies a default device configuration file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. The files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### FileNames

Specifies the name of the file or data set in which to store a backup copy of device configuration information. You can specify more than one file or data set by separating each file name with a comma, with no intervening spaces. This parameter is optional.

If you do not specify a list of files, ADSM stores the information in all files specified with the DEVCONFIG option in the dsmserv.opt file.

### Examples

#### Task

Store a backup copy of the device configuration information in a file named DEVICE.

**Command:** BACKUP DEVCONFIG FileNames=device

## BACKUP DEVCONFIG

### Output:

```
ANR2393I BACKUP DEVCONFIG: Server device configuration
information was written to device.
```

### Related Commands

Table 19 provides references to related commands.

*Table 19. Commands Related to BACKUP DEVCONFIG*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
UPDATE DEVCLASS	Changes the attributes of a device class.	Page 550
UPDATE DRIVE	Changes the attributes of a drive.	Page 575
UPDATE LIBRARY	Changes the attributes of a library.	Page 577

## BACKUP STGPOOL

### BACKUP STGPOOL (Back Up Primary Storage Pool to Copy Storage Pool)

Use the BACKUP STGPOOL command to create backup copies of files that reside in a primary storage pool, and store the backup copies in a copy storage pool. If a file is already duplicated in the specified copy storage pool, a new copy of the file is not made in that copy pool. However, if a copy of a file already exists in the copy storage pool, but the copy is marked *damaged*, a new copy will be created, provided that the primary file is not also marked *damaged*.

If this command is used to back up a random-access storage pool containing cached copies of migrated files or damaged primary files, neither cached copies nor damaged primary files are backed up.

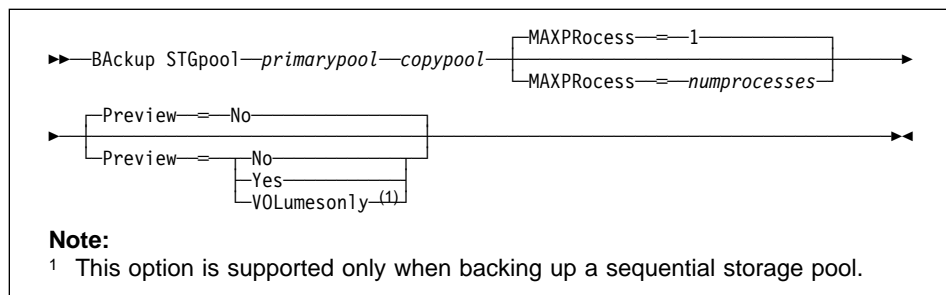
This command generates a background process that can be canceled with the CANCEL PROCESS command. If a BACKUP STGPOOL background process is canceled, some files may have already been backed up prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

Consider the hierarchy you have set up for storage pool migration when planning the backup of storage pools. If migration for a storage pool starts while that storage pool is being backed up, some files may migrate off the storage pool before they are backed up to the copy storage pool. You may want to back up storage pools that are higher in the migration hierarchy before backing up storage pools that are lower. For more information, see *ADSM Administrator's Guide*.

### Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the copy storage pool in which backup copies are to be produced.

### Syntax



### Parameters

*primarypool*

Specifies the name of the primary storage pool that contains the files to be copied. This parameter is required.



## BACKUP STGPOOL

### *copypool*

Specifies the name of the copy storage pool in which the backup copies will be stored. This parameter is required.

### **MAXProcess**=*numprocesses*

Specifies the maximum number of parallel processes to use for backing up files. By using multiple, parallel processes, you may be able to improve throughput for the backup operation. This parameter is optional. You can specify a value from 1–999, inclusive. The default value is 1.

When determining the number of processes to use, consider the number of mount points and drives that can be dedicated to this operation. To access a volume in a sequential access device class, ADSM uses a mount point (a logical drive), and, if the sequential device class is not FILE, a physical drive. The number of mount points and drives available depends on other ADSM and system activity, and on the mount limits of the device classes for the sequential access storage pools that are involved in the backup operation. The maximum number of mount points for a device class is specified with the mount limit parameter, which is the number of logical drives that ADSM can use at one time to access volumes in that device class.

Each process needs a mount point for copy storage pool volumes, and, if the device class is not FILE, each process also needs a drive. If the storage pool you are backing up is a sequential storage pool, each process needs an additional mount point for primary storage pool volumes, and, if the device class is not FILE, an additional drive.

For example, suppose you specify MAXPROCESS=3 to back up a primary sequential storage pool to a copy storage pool, both assigned to the same 8MM device class. Each of the three processes requires two mount points and two drives. For all three processes to run, the 8MM device class must have a mount limit of no less than 6, and there must be at least 6 mount points and 6 drives currently available.

If you specify PREVIEW=YES or PREVIEW=VOLUMESONLY, only one process is used, regardless of the value specified for the MAXPROCESS parameter. No mount points or drives are required when you specify PREVIEW=YES or PREVIEW=VOLUMESONLY.

### **Preview**=*value*

Specifies whether you want to preview the results of the backup operation without actually backing up the data. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the backup operation is to be performed.

#### **Yes**

Specifies that you want to preview the results of the backup operation without actually backing up the data. This option displays the number of files and the number of bytes that will be backed up if you perform the backup. For sequential primary storage pools, this option also produces a

## BACKUP STGPOOL

list of primary storage pool volumes containing files that have not yet been backed up to the specified copy storage pool. These volumes must be mounted if you actually perform the backup.

**Note:** If you only need a list of primary storage pool volumes that must be mounted during backup, execution time for this command can be significantly reduced if you use the VOLUMESONLY option.

### VOLUMESONLY

Specifies that you want to preview the backup operation, but only want to display a list of primary storage pool volumes containing files that have not yet been backed up to the specified copy storage pool. These volumes must be mounted if you actually perform the backup. This option is supported only for backup of sequential primary storage pools.

## Examples

### Task

Back up files in primary storage pool, PRIMARY\_POOL to copy storage pool, COPYSTG.

**Command:** Backup STGpool primary\_pool copystg

### Output:

```
ANR1210I Backup of primary storage pool PRIMARY_POOL to copy
storage pool COPYSTG started as process 16.
ANR1228I Removable volume c:\admserv\mnvol is required
for data storage backup.
ANR8324I FILE volume c:\admserv\mnvol is expected to
be mounted (R/W).
ANR2110I BACKUP STGPOOL started as process 16.
adsm>
ANR8340I FILE volume c:\admserv\mnvol mounted.
ANR8340I FILE volume c:\admserv\data2.dsm mounted.
ANR1212I Backup process 16 ended for storage pool PRIMARY_POOL.
ANR1214I Backup of primary storage pool PRIMARY_POOL
to copy storage pool COPYSTG has ended.
Files Backed Up: 42, Bytes Backed Up: 7452550, Unreadable
Files: 0, Unreadable Bytes: 0.
```

## Related Commands

Table 20 provides references to related commands.

Table 20 (Page 1 of 2). Commands Related to BACKUP STGPOOL

Command	Description	Reference
CANCEL PROCESS	Cancel a background server process.	Page 74

## BACKUP STGPOOL

*Table 20 (Page 2 of 2). Commands Related to BACKUP STGPOOL*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY STGPOOL	Displays information about one or more storage pools.	Page 436
RESTORE STGPOOL	Restores files to a primary storage pool from copy storage pools.	Page 488
RESTORE VOLUME	Restores files stored on specified volumes in a primary storage pool from copy storage pools.	Page 493

## BACKUP VOLHISTORY

---

### BACKUP VOLHISTORY (Save Sequential Volume History Information)

Use the BACKUP VOLHISTORY command to request that the server save sequential volume history information to one or more files.

Volume history information includes data such as date and time of use for the following types of volumes:

- Volumes not in storage pools
  - Volumes used to back up an ADSM database (full or incremental)
  - Volumes used to dump an ADSM database
  - Volumes used to export administrator, node, policy, or server data
- Volumes in storage pools
  - Storage pool sequential access volumes that have been added
  - Storage pool sequential access volumes that have been reused in storage pools through reclamation or MOVE DATA operations
  - Storage pool sequential access volumes that have been removed by using the DELETE VOLUME command or removed during reclamation of scratch volumes

Sequential access volumes are volumes, such as tape or optical volumes, that are accessed sequentially by the ADSM server.

To display sequential volume history information collected by the server, use the QUERY VOLHISTORY command.

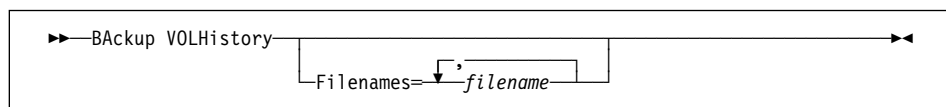
To delete sequential volume history information collected by the server, use the DELETE VOLHISTORY command.

Back up volume history information to external files so that it may be used later to reload server database contents and audit affected storage pool volumes. The external files can be used by the administrator when the server cannot be started to query the database information collected for these volumes. A server options file parameter, VOLUMEHISTORY, can be used to specify one or more files that should automatically be updated whenever server sequential volume history information is changed.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



## BACKUP VOLHISTORY

### Filenames

Specifies the names of one or more files or data sets in which to store a backup copy of volume history information. Separate each file name with a comma and no intervening spaces. This parameter is optional.

If you do not specify a file name, ADSM stores the information in all files specified with the VOLUMEHISTORY option in the **dsmserv.opt** file.

### Examples

#### Task

Back up the volume history information in a file called VOLHIST.

**Command:** BBackup VOLHistory Filenames=volhist

#### Output:

```
ANR2462I BACKUP VOLHISTORY: Server sequential volume
history information was written to volhist.
```

### Related Commands

Table 21 provides references to related commands.

*Table 21. Commands Related to BACKUP VOLHISTORY*

Command	Description	Reference
DELETE VOLHISTORY	Removes sequential volume history information from the volume history file.	Page 234
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
QUERY VOLHISTORY	Displays sequential volume history information that has been collected by the server.	Page 443
UPDATE VOLHISTORY	Adds or changes location information for a volume in the volume history file.	Page 617

## CANCEL PROCESS

---

### CANCEL PROCESS (Cancel an Administrative Process)

Use the CANCEL PROCESS command to cancel a background process that has been started as the result of an administrative command, or from an internal server process (like migration).

To use this command, specify the process number of the process that you want to cancel. If you do not know the process number, obtain it by using the QUERY PROCESS command.

The following commands generate background processes:

AUDIT LIBRARY	EXPORT NODE
AUDIT LICENSES	EXPORT POLICY
AUDIT VOLUME	EXPORT SERVER
BACKUP DB	EXTEND LOG
BACKUP STGPOOL	IMPORT ADMIN
CHECKIN LIBVOLUME	IMPORT NODE
CHECKOUT LIBVOLUME	IMPORT POLICY
DEFINE DBCOPY	IMPORT SERVER
DEFINE LOGCOPY	MOVE DATA
DELETE DBVOLUME	REDUCE DB
DELETE FILESPACE	REDUCE LOG
DELETE LOGVOLUME	RESTORE STGPOOL
DELETE VOLUME	RESTORE VOLUME
EXPIRE INVENTORY	VARY
EXPORT ADMIN	

The following internal server operations generate background server processes:

- Inventory expiration
- Migration
- Reclamation

If the process you want to cancel is currently waiting for a sequential access media volume to be mounted, the mount request is automatically cancelled.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
▶▶—CANce1 PRocess—processnum————▶▶
```

### Parameters

*processnum*

Specifies the number of the background process you want to cancel. This parameter is required.

## CANCEL PROCESS

### Usage Notes

1. If you cancel an internal server process (like migration), that process can restart immediately. For example, if you cancel a migration process, then another migration process for the same storage pool can start if additional data is stored in the storage pool (forcing migration to begin).
2. To permanently cancel a migration process, you can modify the attributes of the storage pool (NEXTSTGPOOL="" or HIGHMIG=100).
3. To permanently cancel a reclamation process, you can modify an attribute of the storage pool (RECLAIM=100).

### Examples

#### Task

Cancel background process number 3.

**Command:** CANCEL PProcess 3

#### Output:

```
ANR0940I Cancel request accepted for process 3.
```

### Related Commands

Table 22 provides a reference to a related command.

*Table 22. Command Related to CANCEL PROCESS*

Command	Description	Reference
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## CANCEL REQUEST

---

### CANCEL REQUEST (Cancel One or More Mount Requests)

Use the CANCEL REQUEST command to cancel one or more pending media mount requests. To cancel a mount request, you need to know the request number assigned to the request. This number is included on the mount request message and can also be shown by using the QUERY REQUEST command.

By specifying the PERMANENT option, you can indicate to the server that the volume is unavailable for further mounts. This causes the server to mark the volume's status as unavailable. Volumes marked *unavailable* are not requested by the server until an UPDATE VOLUME command is issued to change the access mode.

#### Privilege Class

To issue this command, you must have system privilege or operator privilege.

#### Syntax

```
▶▶ CANCEL REQUEST [requestnum] [ALL] [PERMANENT] ▶▶
```

#### Parameters

*requestnum*

Specifies the request number of the mount request to be canceled. You must specify a request number or ALL.

**ALL**

Specifies that all pending mount requests are canceled. You must specify a request number or ALL.

**PERMANENT**

Specifies that you want the server to flag the volumes for which you are canceling a mount request as *unavailable*. This parameter is optional.

#### Examples

##### Task

Cancel request number 2.

**Command:** cancel request 2

##### Output:

```
ANR8346I No requests are outstanding.
```



## CANCEL REQUEST

### Related Commands

Table 23 provides references to related commands.

*Table 23. Commands Related to CANCEL REQUEST*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY REQUEST	Displays information about all pending mount requests.	Page 421
UPDATE VOLUME	Updates the attributes of one or more storage pool volumes.	Page 619

## CANCEL SESSION

---

### CANCEL SESSION (Cancel One or More Client Sessions)

Use the CANCEL SESSION command to cancel an existing administrative or client node session. You can use this command to force an administrative or client node session off of ADSM. To prevent any future client node sessions from beginning, use the DISABLE command.

When you issue this command, any outstanding mount requests related to this session are automatically canceled.

If you cancel an administrative or client node session that is in the *idle wait* (IdleW) state, the client session automatically reconnects the next time that an action is performed.

To prevent a specific administrator or client node from accessing ADSM, use the LOCK ADMIN or LOCK NODE command. If the client is currently accessing the server, the client session must end or be canceled before that node can be locked out with the LOCK NODE command.

You must cancel administrative client sessions by session number. Use the QUERY SESSION command to display information on all sessions that can be canceled, including session numbers.

When you cancel a client node session, the client node must start a new session to resume ADSM activities. If a process, such as backup or archive, is interrupted by this command, the results of any processing up to the point of interruption are rolled back and not committed to the database.

When you cancel a client node session, the client node must start a new session to resume ADSM activities from the point of interruption. The results of any incomplete processing at the point of interruption are rolled back and not committed to the database. Processing that is complete prior to the point of interruption, however, is committed and is not rolled back.

### Privilege Class

To issue this command, you must have system privilege or operator privilege.

### Syntax

```
▶▶ CANCEL SESSION [sessionnum] [ALL] ▶▶
```

### Parameters

*sessionnum*

Specifies the number of the administrative or client node session that you want to cancel. You must specify a session number or ALL.

## CANCEL SESSION

### ALI

Specifies that all client node sessions are canceled. You must specify a session number or ALL.

You cannot use this parameter to cancel administrative client sessions.

## Examples

### Task 1

Cancel the client node session with NODEP (session 3).

**Command:** CAnceL SEssion 3

**Output:**

```
ANR0490I Canceling session 3.
```

### Task 2

Cancel the administrative session with MARIE (session 5).

**Command:** CAnceL SEssion 5

**Output:**

```
ANR0490I Canceling session 5.
```

## Related Commands

Table 24 provides references to related commands.

*Table 24. Commands Related to CANCEL SESSION*

Command	Description	Reference
DISABLE	Prevents new clients from accessing ADSM but permits existing sessions to continue.	Page 78
LOCK ADMIN	Prevents an administrator from accessing ADSM.	Page 303
LOCK NODE	Prevents a client from accessing the server.	Page 304
QUERY SESSION	Displays information about all active administrator and client sessions with ADSM.	Page 427

## CHECKIN LIBVOLUME

### CHECKIN LIBVOLUME (Check a Storage Volume into a Library)

Use the CHECKIN LIBVOLUME command to enter a sequential access storage volume into an existing automated library. The CHECKIN LIBVOLUME command informs the ADSM server that a volume is available for use. The server does not use any volumes that physically reside in an automated library until a volume has been checked in via the CHECKIN LIBVOLUME command.

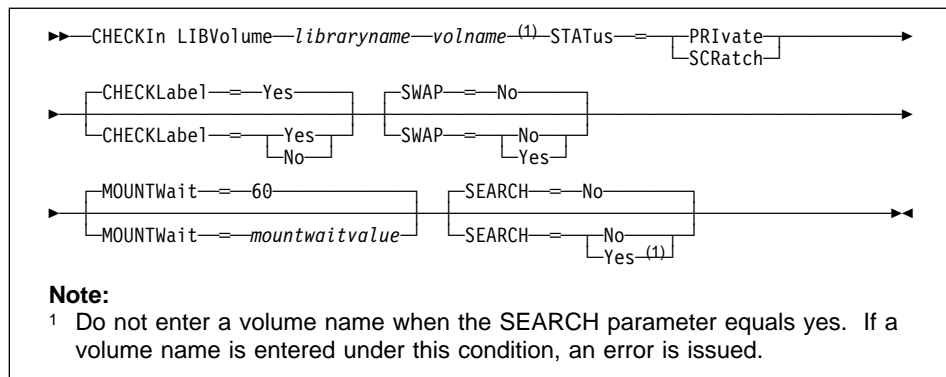
If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command.

This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### *libraryname*

Specifies the name of the library where the storage volume is to be located. This parameter is required.

#### *volname*

Specifies the volume name of the storage volume being checked in. Do not enter this parameter if the SEARCH parameter equals yes. This parameter is required under any other condition.

#### **STATus=voltype**

Specifies the status that is to be assigned to the storage volume being checked in. This keyword is required. Possible values are:

## CHECKIN LIBVOLUME

### **PRivate**

Specifies that the volume is not part of the library's pool of scratch volumes. A PRIVATE volume is only mounted when its volume name is explicitly requested.

### **SCRatch**

Specifies that the storage volume is a new scratch volume. This new scratch volume can be mounted to satisfy scratch mount requests during either data storage operations or export operations.

### **CHECKLabel=checkvalue**

Specifies how or whether ADSM should read sequential media labels or bar code information of volumes during CHECKIN LIBVOLUME command processing. This parameter is optional. Setting the *checkvalue* to NO allows ADSM to utilize, if available, any barcode information. The default value is YES.

Possible values are:

#### **Yes**

Specifies that during check-in an attempt is made to read the media label.

#### **No**

Specifies to use any available barcode information. If barcode information is unavailable, an attempt is made to read the media label.

### **SWAP=swapvalue**

Specifies whether ADSM will initiate a swap operation when an empty library slot is not available during CHECKIN LIBVOLUME command processing. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that if an empty slot is not available, the CHECKIN LIBVOLUME command is rejected.

#### **Yes**

Specifies that if an empty slot is not available, ADSM initiates a swap operation. The volume selected (target swap volume) for the swap operation is ejected from the library and replaced with the volume being checked in. ADSM determines which volume is the target swap volume by checking for (1) any available scratch volumes, and if this is not successful, then (2) the least frequently mounted volume.

You can override swap processing by selecting a volume in the library and then explicitly issuing the CHECKOUT LIBVOLUME command prior to the CHECKIN LIBVOLUME command.

### **MOUNTWait=mountwaitvalue**

Specifies the maximum length of time, in minutes, to wait for a storage volume to be mounted. This parameter is optional. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## CHECKIN LIBVOLUME

### SEARCH=*searchvalue*

Specifies whether ADSM searches the library to see if the volume has been previously checked in. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that only the named volume is to be checked into the library. The server issues a mount request to have the volume inserted into a cartridge slot within the library or if available, inserted into the library's entry/exit port. If the cartridge slot is used, it is identified by its element address, see *ADSM Quick Start* for details on element addresses.

#### Yes

Specifies that ADSM should search the library for volumes that can be checked in automatically. Because volume names are determined from the media label of volumes that already reside in the library, the volume name (*volname*) parameter must be omitted when this option is used.

If the library is shared between applications, be cautious when selecting this parameter. ADSM could examine a volume required by another application.

For SCSI libraries, the server scans all of the slots in the library for volumes that have not previously been checked in. If a volume meets this requirement and contains a valid volume label, it is checked in automatically.

## Examples

### Task

Check in the volume named WPDV00 into the library named AUTO.

**Command:** CHECKIn LIBVolume auto wpdv00 STATus=SCRatch

### Output:

```
ANR8422I CHECKIN LIBVOLUME: Operation for library AUTO
started as process 12.
```

## Related Commands

Table 25 provides references to related commands.

Table 25 (Page 1 of 2). Commands Related to CHECKIN LIBVOLUME

Command	Description	Reference
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51

## CHECKIN LIBVOLUME

*Table 25 (Page 2 of 2). Commands Related to CHECKIN LIBVOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CHECKOUT LIBVOLUME	Checks a storage volume out of an automated library.	Page 84
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
QUERY LIBVOLUME	Displays information about a library volume.	Page 384
UPDATE LIBVOLUME	Changes the status of a storage volume.	Page 579

## CHECKOUT LIBVOLUME

### CHECKOUT LIBVOLUME (Check a Storage Volume out of a Library)

Use the CHECKOUT LIBVOLUME command to remove a sequential access storage volume from an existing automated library.

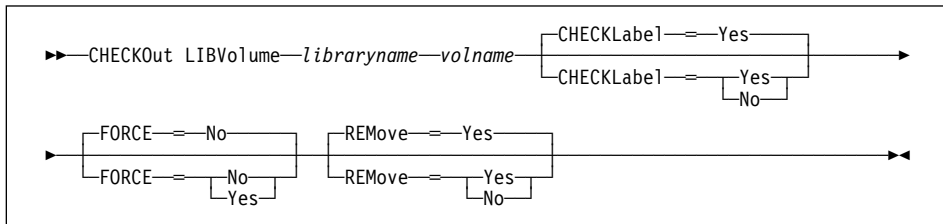
If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command.

This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### *libraryname*

Specifies the name of the library where the storage volume is to be removed. This parameter is required.

#### *volname*

Specifies the volume name of the storage volume being checked out. This parameter is required.

#### **CHECKLabel**=*checkvalue*

Specifies how or whether ADSM should read sequential media labels or bar code information of volumes during CHECKOUT LIBVOLUME command processing. This parameter is optional. Setting the *checkvalue* to NO allows ADSM to utilize, if available, any bar code information. The default value is YES. Possible values are:

#### **Yes**

Specifies that during checkout libvolume processing an attempt is made to read the media label. Reading the media label verifies that the correct volume is being checked out.

#### **No**

Specifies to use any available bar code information.



## CHECKOUT LIBVOLUME

### **FORCE**=*forcevalue*

Specifies whether ADSM checks out a storage volume if there is an input/output (I/O) error reading the label. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that ADSM does not check out a storage volume when there is an I/O error reading the label.

#### **Yes**

Specifies that ADSM performs the storage volume check out even when an I/O error reading the label occurs.

### **REMove**=*removevalue*

Specifies whether ADSM ejects a volume from an SCSI library during CHECKOUT LIBVOLUME command processing. This parameter is optional. The default value is Yes. Possible values are:

#### **Yes**

Specifies that a volume is ejected during CHECKOUT LIBVOLUME command processing. The volume entry is deleted from the ADSM library inventory.

#### **No**

Specifies that a volume is not ejected during CHECKOUT LIBVOLUME command processing. This deletes the volume entry from the ADSM library inventory and leaves the volume in the library for use by other applications.

## Usage Notes

A volume that is checked out of a library may still be defined in a storage pool. If the server attempts to access the checked out volume, a request is issued that waits for the volume to be checked in.

## Examples

### Task

Check out the volume named EXB004 from the library named FOREST.

**Command:** CHECKOut LIBVolume forest exb004

## CHECKOUT LIBVOLUME

### Output:

```
ANS5104I Process number 5 started.

ANR8434I CHECKOUT LIBVOLUME: Operation for volume EXB004
in library FOREST started as process 26.
ANR8336I Verifying label of 8MM volume EXB004 in drive
DRIVE2 (tape01$).
ANR8307I 003: Remove 8MM volume EXB004 from slot 2 of
library FOREST; REPLY when ready.
ANR8438I CHECKOUT LIBVOLUME for volume EXB004 in library
FOREST completed successfully.
```

### Related Commands

Table 26 provides references to related commands.

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
CANCEL PROCESS	Cancels a background server process.	Page 74
CHECKIN LIBVOLUME	Checks a storage volume into an automated library.	Page 80
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
QUERY LIBVOLUME	Displays information about a library volume.	Page 384
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
UPDATE LIBVOLUME	Changes the status of a storage volume.	Page 579

## COMMIT

---

### COMMIT (Control Committing of Commands in a Macro)

Use the COMMIT command in a macro to permanently update the database with all changes that result from the successful completion of any commands in the macro processed up to that point. This command does not generate a message when it is issued from the console mode of the administrative client.

You can control precisely when commands are committed with the COMMIT command. If an error occurs while processing the commands in a macro, the server terminates processing of the macro and rolls back any uncommitted changes (commands that have been processed since the last COMMIT).

Make sure that your administrative client session is not running with the ITEMCOMMIT option if you want to control command processing with this command. After a command is committed, it cannot be rolled back.

For more information on controlling command processing in a macro, see “Controlling Command Processing in a Macro” on page 25.

### Privilege Class

Any administrator can issue this command.

### Syntax

```
▶▶—COMMIT—————▶▶
```

### Parameters

None.

### Examples

#### Task

From the interactive mode of the administrative client, register and grant authority to new administrators using a macro named REG.ADM. Changes are committed after each administrator is registered and has been granted authority.

#### Macro Contents:

```
/* REG.ADM-register policy admin & grant authority*/  
REGister Admin sara hobby  
GRant AUTHority sara Classes=Policy  
COMMIT /* Commits changes */  
REGister Admin ken plane  
GRant AUTHority ken Classes=Policy  
COMMIT /* Commits changes */
```

## COMMIT

**Command:** MACRO reg.adm

**Output:**

```
ANS5101I Server command: 'REGister Admin sara hobby'  
ANR2068I Administrator SARA registered.  
  
ANS5101I Server command: 'GRant AUTHority sara  
Classes=policy'  
ANR2077I Unrestricted policy privilege granted to  
administrator SARA.  
  
ANS5101I Server command: 'COMMIT'  
  
ANS5101I Server command: 'REGister Admin ken plane'  
ANR2068I Administrator KEN registered.  
  
ANS5101I Server command: 'GRant AUTHority ken Classes=policy'  
ANR2077I Unrestricted policy privilege granted to  
administrator KEN.  
  
ANS5101I Server command: 'COMMIT'
```

## Related Commands

Table 27 provides references to related commands.

*Table 27. Commands Related to COMMIT*

Command	Description	Reference
MACRO	Runs a specified macro file.	Page 306
ROLLBACK	Discards any uncommitted changes to the database since the last COMMIT was executed.	Page 503

## COPY DOMAIN

---

### COPY DOMAIN (Copy a Policy Domain)

Use the COPY DOMAIN command to create a copy of an existing policy domain.

When you use this command, ADSM copies the following information:

- Policy domain description
- Policy sets in the policy domain (including the ACTIVE policy set, if defined)
- Management classes in each policy set (including the default management class, if assigned)
- Copy groups in each management class

Use the UPDATE DOMAIN command to modify an existing policy domain. DEFINE, DELETE, and UPDATE commands are available for modifying management classes and copy groups.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—COPy D0main—domainname1—domainname2—◄◄
```

### Parameters

*domainname1*

Specifies the name of the existing policy domain you want to copy. This parameter is required.

*domainname2*

Specifies the name of the new policy domain. This parameter is required. The maximum length of this name is 30 characters.

### Examples

#### Task

Copy the existing policy domain PROG1 to PROG2.

**Command:** COPy D0main prog1 prog2

**Output:**

```
ANR1503I Policy domain PROG1 copied to domain PROG2.
```

## COPY DOMAIN

### Related Commands

Table 28 provides references to related commands.

Table 28 (Page 1 of 2). Commands Related to COPY DOMAIN

Command	Description	Reference
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE COPYGROUP	Deletes a backup or archive copy group from a policy domain and policy set.	Page 199
DELETE DOMAIN	Deletes a policy domain along with any policy objects in the policy domain.	Page 208
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397

## COPY DOMAIN

Table 28 (Page 2 of 2). Commands Related to COPY DOMAIN

Command	Description	Reference
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539
UPDATE DOMAIN	Changes the attributes of a policy domain.	Page 573
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## COPY MGMTCLASS

---

### COPY MGMTCLASS (Copy a Management Class)

Use the COPY MGMTCLASS command to create a copy of an existing management class within the same policy set.

When you use this command, ADSM copies the following information:

- Management class description
- Copy groups defined to the management class
- The following space management attributes, if they are specified in the management class:
  - SPACEMGTECHNIQUE
  - AUTOMIGNONUSE
  - MIGREQUIRESBKUP
  - MIGDESTINATION

After you copy a management class, you can use the UPDATE MGMTCLASS command to change the management class description.

You cannot copy management classes across policy domains or policy sets.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the new management class belongs.

### Syntax

```
►►—COpy MGmtclass—domainname—setname—classname1—classname2—◄◄
```

### Parameters

*domainname*

Specifies the name of the policy domain to which the management class belongs. This parameter is required.

*setname*

Specifies the name of the policy set to which the management class belongs. This parameter is required.

*classname1*

Specifies the name of the existing management class you want to copy. This parameter is required.

*classname2*

Specifies the name of the new management class. This parameter is required. The maximum length of this name is 30 characters.



## COPY MGMTCLASS

### Examples

#### Task

Copy the existing management class ACTIVEFILES to FILEHISTORY for policy set VACATION in the EMPLOYEE\_RECORDS policy domain.

**Command:** `COPY MGMTCLASS employee_records vacation activefiles filehistory`

#### Output:

```
ANR1523I Management class ACTIVEFILES copied to
class FILEHISTORY in policy domain EMPLOYEE_RECORDS,
set VACATION.
```

### Related Commands

Table 29 provides references to related commands.

*Table 29. Commands Related to COPY MGMTCLASS*

Command	Description	Reference
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581

## COPY POLICYSET

---

### COPY POLICYSET (Copy a Policy Set)

Use the COPY POLICYSET command to create a copy of any existing policy set (including the ACTIVE policy set) within the same policy domain.

When you use this command, ADSM copies the following information:

- Policy set description
- Management classes in the policy set (including the default management class, if assigned)
- Copy groups in each management class

A policy set cannot be copied across policy domains.

To change the description of the new policy set, use the UPDATE POLICYSET command.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the new policy set belongs.

### Syntax

```
►► COPY POLICYSET domainname setname1 setname2 ◀◀
```

### Parameters

*domainname*

Specifies the name of the policy domain to which the policy set belongs. This parameter is required.

*setname1*

Specifies the name of the existing policy set you want to copy. This parameter is required.

*setname2*

Specifies the name of the new policy set. This parameter is required. The maximum length of this name is 30 characters.

### Examples

#### Task

Copy the existing policy set VACATION to HOLIDAY in the EMPLOYEE\_RECORDS policy domain.

## COPY POLICYSET

**Command:** COpY P0licyset employee\_records vacation holiday

**Output:**

```
ANR1513I Policy set VACATION copied to set HOLIDAY in
policy domain EMPLOYEE_RECORDS.
```

### Related Commands

Table 30 provides references to related commands.

*Table 30. Commands Related to COPY POLICYSET*

Command	Description	Reference
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## COPY SCHEDULE

---

### COPY SCHEDULE (Copy a Client or an Administrative Command Schedule)

Use the COPY SCHEDULE command to create a copy of an existing schedule. The COPY SCHEDULE command takes two forms, depending on whether the schedule applies to client operations or administrative commands. The syntax and parameters for each form are defined separately.

---

#### Copying Client Schedules

Use the COPY SCHEDULE command to create a copy of an existing schedule for client operations. You can copy a schedule within a policy domain or from one policy domain to another existing policy domain.

When you use this command, ADSM copies the following information:

- Description of the schedule
- All parameter values in the original schedule

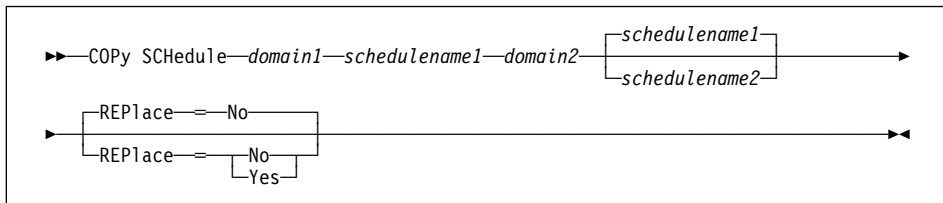
The last modification value for the new schedule is set to the time that you executed this command. Any clients associated with the original schedule are *not* associated with the new schedule. An administrator with policy privilege must use the DEFINE ASSOCIATION command to associate clients to the new schedule.

To make changes to an existing schedule, use the UPDATE SCHEDULE command.

#### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which you are copying the schedule.

#### Syntax for Copying Client Schedules



#### Parameters for Copying Client Schedules

*domain1*

Specifies the name of the policy domain that contains the schedule you want to copy. This parameter is required.

*schedulename1*

Specifies the name of the schedule you want to copy. This parameter is required.

## COPY SCHEDULE

### *domain2*

Specifies the name of an existing policy domain to which you want to copy the new schedule. This parameter is required. You must have policy privilege for this policy domain to use this command. You must specify a new schedule name (with the SCHEDULENAME2 parameter) if you want to copy a schedule within a policy domain.

### *schedulename2*

Specifies the name of the new schedule. This parameter is optional. The maximum length of the name is 30 characters.

If you do not specify this name, the name of the original schedule (*schedulename1*) is used.

If the schedule name is already defined in the policy domain, you must specify REPLACE=YES, or the command fails.

### **REPlace=replacevalue**

Specifies whether an existing schedule can be replaced. This parameter is optional. The default value is NO.

If the new schedule name is already defined in the new policy domain, you must specify REPLACE=YES, or the command fails. Possible values are:

#### **No**

Specifies that an existing schedule cannot be replaced.

#### **Yes**

Specifies that an existing schedule can be replaced.

---

## Copying Administrative Command Schedules

Use the COPY SCHEDULE command to create a copy of an existing administrative command schedule. You can copy an administrative command schedule to another administrative command schedule.

When you use this command, ADSM copies the following information:

- Description of the schedule
- All parameter values in the original schedule

The last modification value for the new schedule is set to the time that you executed this command.

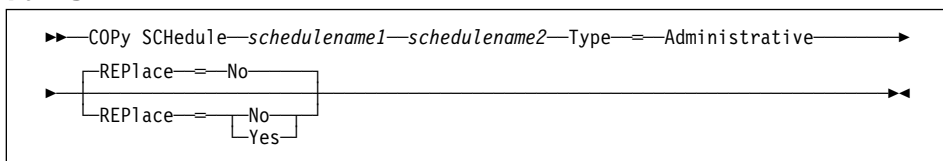
To make changes to an existing schedule, use the UPDATE SCHEDULE command.

## Privilege Class

To issue this command, you must have system privilege.

## COPY SCHEDULE

### Syntax for Copying Administrative Command Schedules



### Parameters for Copying Administrative Command Schedules

#### *schedulename1*

Specifies the name of the schedule you want to copy. This parameter is required.

#### *schedulename2*

Specifies the name of the new schedule. This parameter is required. The maximum length of the name is 30 characters.

If the schedule name is already defined, you must specify REPLACE=YES, or the command fails.

#### **Type=Administrative**

Specifies this is a scheduled administrative command. This parameter is required.

#### **REPLACE=replacevalue**

Specifies whether an existing schedule can be replaced. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that an existing schedule cannot be replaced.

#### **Yes**

Specifies that an existing schedule can be replaced.

## Examples

### Task 1

Copy the WEEKLY\_BACKUP schedule that belongs to policy domain EMPLOYEE\_RECORDS to the PROG1 policy domain and name the new schedule WEEKLY\_BACK2. If there is already a schedule with this name defined in the PROG1 policy domain, do not replace it.

**Command:** copy schedule employee\_records weekly\_backup prog1 weekly\_back2

#### **Output:**

```
ANR2503I Schedule WEEKLY_BACKUP in domain EMPLOYEE_RECORDS copied to  
schedule WEEKLY_BACK2 in domain PROG1.
```

## COPY SCHEDULE

### Task 2

Copy the DAILYBACKUP schedule that is defined to the EMPLOYEE\_RECORDS policy domain to the PROG1 policy domain. If there is already a DAILYBACKUP schedule defined to the PROG1 policy domain, it is replaced.

**Command:** `copy schedule employee_records dailybackup prog1 dailybackup replace=yes`

#### Output:

```
ANR2503I Schedule DAILYBACKUP in domain EMPLOYEE_RECORDS
copied to schedule DAILYBACKUP in domain PROG1.
```

### Task 3

Copy existing administrative schedule BACKUP\_ARCHIVEPOOL to a new schedule named BCKSCHED.

**Command:** `copy schedule backup_archivepool bcksched type=administrative`

#### Output:

```
ANR2513I Schedule BACKUP_ARCHIVEPOOL copied to schedule
BCKSCHED.
```

## Related Commands

Table 31 provides references to related commands.

*Table 31. Commands Related to COPY SCHEDULE*

Command	Description	Reference
DEFINE ASSOCIATION	Associates one or more clients with a schedule.	Page 100
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164
DELETE SCHEDULE	Deletes a schedule from the database.	Page 229
QUERY SCHEDULE	Displays information about one or more schedules.	Page 422
UPDATE SCHEDULE	Changes the attributes of a schedule.	Page 590

## DEFINE ASSOCIATION

---

### DEFINE ASSOCIATION (Associate Client Nodes with a Schedule)

Use the DEFINE ASSOCIATION command to associate one or more clients with a schedule. Client nodes that are associated with a schedule initiate backup and archive functions according to that schedule.

**Note:** In a macro, the server may stall if some commands (for example, REGISTER NODE and DEFINE ASSOCIATION) are not committed as soon as you issue them. You could follow each command in a macro with a COMMIT command. However, a simpler solution is to include the -ITEMCOMMIT option with the DSMADMC command.

Before associating a client to a schedule, the following conditions must be true:

- The specified schedule exists
- The client is registered to the policy domain to which the schedule applies

Any clients already associated with a specified schedule are not affected when you associate new clients with that schedule.

A client remains associated with a schedule until you use the DELETE ASSOCIATION command to remove it from the list of associated clients or until the schedule is deleted by using the DELETE SCHEDULE command.

In addition, any clients that you subsequently define to ADSM are *not* automatically associated with a schedule. You must use the DEFINE ASSOCIATION command to associate any newly registered clients with the appropriate schedules.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the schedule belongs.

### Syntax

```
►►—DEfIne ASSOCIation—domainname—schedulename—nodename—◄◄
```

### Parameters

#### *domainname*

Specifies the name of the policy domain to which the schedule belongs. This parameter is required.

#### *schedulename*

Specifies the name of the schedule that you want to associate with one or more clients. This parameter is required.

#### *nodename*

Specifies the name of the client node to be associated with the specified schedule. This parameter is required. You can specify a list of clients that you want to associate with the specified schedule. The items in the list are separated



## DEFINE ASSOCIATION

by commas, with no intervening spaces. You can use a pattern-matching expression to specify a name. All matching clients are associated with the specified schedule. If a client is listed, but is already associated with the specified schedule or is not assigned to the domain to which the schedule belongs, the command has no effect for that client.

### Examples

#### Task 1

Associate clients with the WEEKLY\_BACKUP schedule. The associated clients have names that begin with SMITH or JOHN, and are assigned to the EMPLOYEE\_RECORDS policy domain.

**Command:** `DEFine ASSOCiation employee_records weekly_backup smith*,john *`

#### Output:

```
ANR2510I Node JOHN04 associated with schedule WEEKLY_BACKUP
in policy domain EMPLOYEE_RECORDS.
ANR2510I Node JOHNNY associated with schedule WEEKLY_BACKUP
in policy domain EMPLOYEE_RECORDS.
ANR2510I Node SMITH associated with schedule WEEKLY_BACKUP
in policy domain EMPLOYEE_RECORDS.
```

#### Task 2

Associate the clients JOE, TOM, and LARRY with the WINTER schedule, assigned to the EMPLOYEE\_RECORDS policy domain. However, the client JOE is already associated with the WINTER SCHEDULE.

**Command:** `DEFine ASSOCiation employee_records winter joe,tom,larry`

#### Output:

```
ANR2510I Node TOM associated with schedule WINTER in policy
domain EMPLOYEE_RECORDS.
ANR2510I Node LARRY associated with schedule WINTER in policy
domain EMPLOYEE_RECORDS.
```

### Related Commands

Table 32 provides references to related commands.

Table 32 (Page 1 of 2). Commands Related to DEFINE ASSOCIATION

Command	Description	Reference
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164

## DEFINE ASSOCIATION

*Table 32 (Page 2 of 2). Commands Related to DEFINE ASSOCIATION*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DELETE ASSOCIATION	Deletes the association between one or more clients and a schedule.	Page 197
DELETE SCHEDULE	Deletes a schedule from the database.	Page 229
QUERY ASSOCIATION	Displays the clients associated with one or more schedules.	Page 322
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466

## DEFINE COPYGROUP

---

### DEFINE COPYGROUP (Define a Copy Group)

Use the DEFINE COPYGROUP command to define a new backup or archive copy group within a specified policy domain, policy set, and management class. This command also defines the destination storage pool and backup or archive attributes for the copy group. When you define a copy group, it must be named STANDARD.

**Note:** The DEFINE COPYGROUP will fail if a copy storage pool name is specified as a destination.

A *backup copy group* contains attributes that ADSM uses during the backup process to determine the following:

- Whether a file is only backed up when the file has changed since the last time it was backed up
- Whether a file that is in use is backed up
- How many days must elapse before a file can again be backed up
- How to process a file that is in use during backup

A backup copy group also contains attributes that ADSM uses to manage the backup versions of your files. These attributes control the following:

- Where the server stores backup versions of client node files
- How many backup versions the server keeps of client node files and directories
- How long the server keeps backup versions of client node files and directories

An *archive copy group* contains attributes that ADSM uses during the archive process to determine the following:

- Whether a file that is in use is archived
- Where the server stores archived copies of client node files
- How long the server keeps archived copies of client node files

You can define one backup and one archive copy group for each management class. If you plan to have the client nodes in your installation perform backups, it is recommended that the default management class contain a backup copy group.

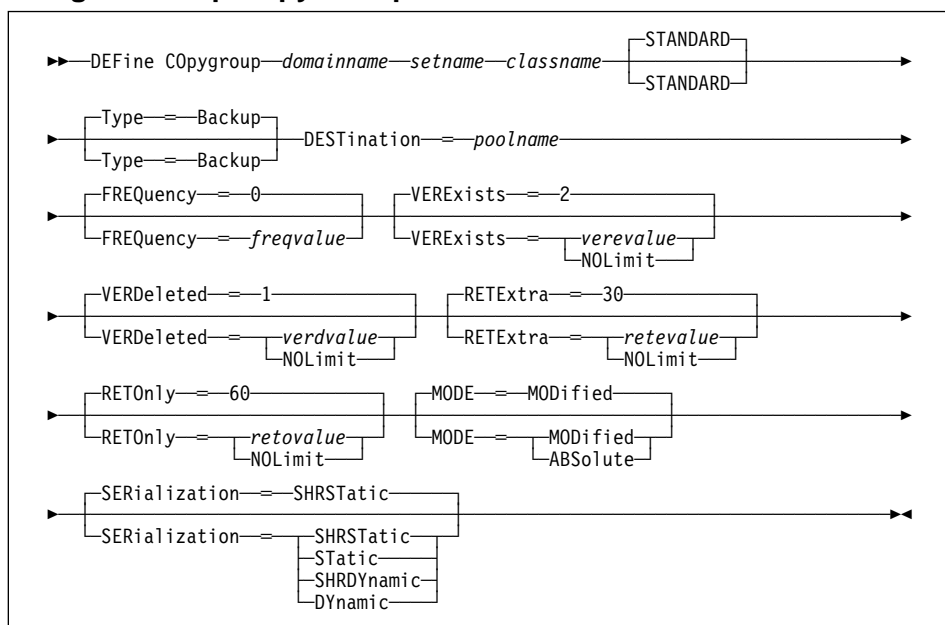
**Note:** Since there are two types of copy groups (backup and archive), the DEFINE COPYGROUP command syntax diagram and parameter listing is displayed twice—once for defining a backup copy group and once for defining an archive copy group.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the copy group belongs.

## DEFINE COPYGROUP

### Syntax for Defining a Backup Copy Group



### Parameters for Defining a Backup Copy Group

#### *domainname*

Specifies the name of the policy domain for which you are defining the copy group. This parameter is required.

#### *setname*

Specifies the name of the policy set for which you are defining the copy group. This parameter is required.

You cannot define a copy group for a management class that belongs to the ACTIVE policy set.

#### *classname*

Specifies the name of the management class for which you are defining the copy group. This parameter is required.

#### **STANDARD**

Specifies the name of the copy group as STANDARD. The name of the copy group must be STANDARD. This parameter is optional. The default value is STANDARD.

#### **Type=Backup**

Specifies that you want to define a backup copy group. The default parameter is BACKUP.

## DEFINE COPYGROUP

### **DESTination**=*poolname*

Specifies the name of the primary storage pool where backup data is to be stored. This parameter is required. You can only specify one storage pool as the destination for a copy group.

### **FREQuency**=*freqvalue*

Specifies the minimum interval, in days, between successive backups. This parameter is optional. ADSM does not perform an incremental backup unless the specified number of days has elapsed since the last backup. The FREQUENCY value is only used during incremental backup. This value is ignored during selective backup. You can specify an integer from 0–9999, inclusive. The default value is 0.

### **VERExists**=*vervalue*

Specifies the maximum number of backup versions to retain for files that are currently on the client file system. This parameter is optional.

If a backup causes the limit to be exceeded, the oldest backup version that exists is deleted from the storage pool. The default value is 2. Possible values are:

#### *vervalue*

Specifies the number of backup versions to retain for files that are currently on the client file system. You can specify an integer from 1–9999, inclusive. The default value is 2.

### **NOLimit**

Specifies that you want ADSM to retain all backup versions.

### **VERDeleted**=*verdvalue*

Specifies the maximum number of backup versions to retain for files that are deleted from the client file system. This parameter is optional. This parameter is ignored if the file still resides at the client node.

If the client file is deleted, the next backup causes the oldest versions in excess of this number to be deleted from the backup storage pool. The expiration date for the remaining versions is determined by the values specified for the RETEXTRA and RETONLY parameters. The default value is 1. Possible values are:

#### *verdvalue*

Specifies an integer that defines the number of backup versions to retain for files that are deleted from the client file system. You can specify an integer from 0–9999, inclusive.

### **NOLimit**

Specifies that you want ADSM to retain all backup versions for files that are deleted from the client file system.

### **RETEExtra**=*retevalue*

Specifies the retention time, in days, for all but the most recent backup version. This parameter is optional. The default value is 30 days. Possible values are:

## DEFINE COPYGROUP

### *retevalue*

Specifies an integer that defines the number of days to retain all backup versions other than the most recent version. You can specify an integer from 0–9999, inclusive.

### **NOLimit**

Specifies that you want ADSM to retain all backup versions, other than the most recent version, indefinitely. The most recent version of a file that exists on the client workstation is retained indefinitely or until the user deletes the file on the workstation. If the user deletes the file on the workstation, the file is rolled off by using the value specified for the VERDELETED parameter.

If the NOLIMIT value is specified, extra backup versions are deleted under control of the VEREXISTS or VERDELETED parameters.

### **RETOOnly**=*retovalue*

Specifies the retention time, in days, for the last backup version of a file that has been deleted from the client's file system. This parameter is optional. The default value is 60. Possible values are:

### *retovalue*

Specifies an integer that defines the number of days to retain the last remaining inactive version of a file. You can specify an integer from 0–9999, inclusive.

### **NOLimit**

Specifies that the last remaining inactive version of a file be kept indefinitely.

### **MODE**=*mode*

Specifies whether a file should be backed up based on changes made to the file since the last time it was backed up. This parameter is optional. The MODE value is only used for incremental backup. This value is ignored during selective backup. The default value is MODIFIED. Possible values are:

### **MODified**

Specifies that the file is backed up only if it has changed since the last backup.

### **ABSolute**

Specifies that the file is backed up regardless of whether it has been modified.

### **SERialization**=*serialvalue*

Specifies how files or directories are handled if they are modified during backup processing and what ADSM should do if a modification occurs. This parameter is optional. The default value is SHRSTATIC.

**Attention:** Be careful about choosing a management class that contains a copy group that specifies shared dynamic (SHRDYNAMIC) or DYNAMIC for SERIALIZATION.

## DEFINE COPYGROUP

If you choose a management class that allows a file or directory to be backed up while modifications are occurring, the backup version stored on the server might be a fuzzy backup. A fuzzy backup is a backup version that does not accurately reflect what is currently in the file or directory. The fuzzy backup contains some, but not all, modifications. If that is not acceptable, be sure to choose a management class that allows ADSM to create a backup version only if the file or directory is not being modified.

If a file is restored that contains a fuzzy backup, the file may not be usable, depending on the file's application.

Possible values are:

### **SHRStatic**

Specifies that a file or directory must not be backed up while being modified. ADSM attempts to perform a backup as many as four times, depending on the value specified for CHANGINGRETRIES in the client user options file. If the file or directory is modified during each backup attempt, ADSM does not back it up.

### **Static**

Specifies that a file or directory must not be modified during backup. If a file or directory is modified during the backup attempt, it is not backed up.

Platforms that do not support the STATIC option default to SHRSTATIC. The STATIC option is not supported on the DOS platform.

### **SHRDynamic**

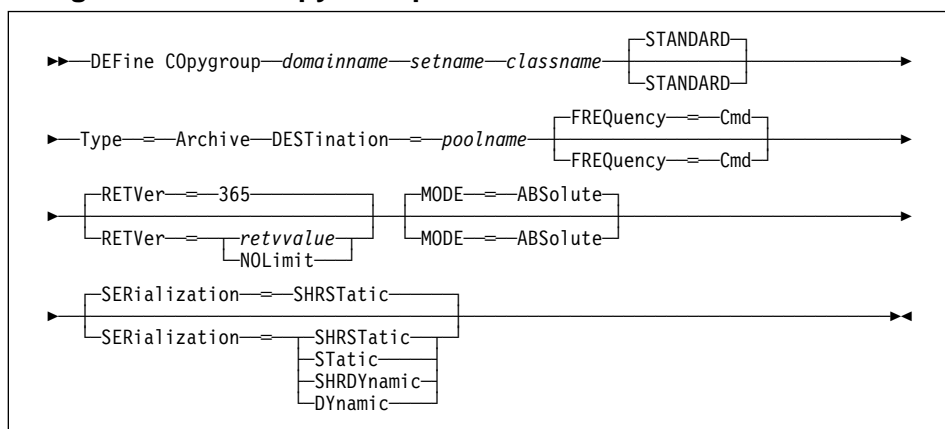
Specifies that if the file or directory is being modified during a backup attempt, ADSM backs up the file or directory during the last attempt even though the file or directory is being modified. ADSM attempts to perform a backup as many as four times, depending on the value specified for CHANGINGRETRIES in the client user options file.

### **DYnamic**

Specifies that a file or directory is backed up on the first attempt by ADSM, regardless of whether the file or directory is being modified during backup processing.

## DEFINE COPYGROUP

### Syntax for Defining an Archive Copy Group



### Parameters for Defining an Archive Copy Group

#### *domainname*

Specifies the name of the policy domain for which you are defining the copy group. This parameter is required.

#### *setname*

Specifies the name of the policy set for which you are defining the copy group. This parameter is required.

#### *classname*

Specifies the name of the management class for which you are defining the copy group. This parameter is required.

#### **STANDARD**

Specifies the name of the copy group. The name of the copy group must be STANDARD. This parameter is optional. The default value is STANDARD.

#### **Type=Archive**

Specifies that you want to define an archive copy group. This parameter is required.

#### **DESTINATION=poolname**

Specifies the name of the primary storage pool where the archive files are to reside. This parameter is required. You can only specify one storage pool as the destination for a copy group.

#### **FREQUENCY=Cmd**

Specifies the value for the copy frequency. The copy frequency must be CMD. This parameter is optional. The default value is CMD.

#### **RETVER=retvvalue**

Specifies the number of days to keep an archive copy bound to this management class. This parameter is optional. Possible values are:



## DEFINE COPYGROUP

### *retvalue*

Specifies an integer that defines the number of days to retain an archive copy. You can specify an integer from 0–9999, inclusive. The default value is 365.

### **NOLimit**

Specifies that an archive copy is maintained indefinitely (that is, inventory expiration processing does not affect the archive copy).

### **MODE=ABSolute**

Specifies that a file is archived. The file must be archived as ABSOLUTE. This parameter is optional. The default value is ABSOLUTE.

### **SERialization=serialvalue**

Specifies how files are handled if they are modified during archive processing and what ADSM should do if a modification occurs. This parameter is optional. The default value is SHRSTATIC.

**Attention:** Be careful about choosing a management class that contains a copy group that specifies shared dynamic (SHRDYNAMIC) or DYNAMIC for SERIALIZATION.

If you choose a management class that allows a file to be archived while modifications are occurring, the archive copy stored on the server might be a fuzzy copy. A fuzzy copy is an archive copy that does not accurately reflect what is currently in the file. The fuzzy copy contains some, but not all, modifications. If that is not acceptable, be sure to choose a management class that allows ADSM to create an archive copy only if the file is not being modified.

If a file is restored that contains a fuzzy copy, the file may not be usable, depending on the file's application.

Possible values are:

### **SHRStatic**

Specifies that a file must not be archived while being modified. ADSM attempts to perform an archive as many as four times, depending on the value specified for CHANGINGRETRIES in the client user options file. If the file is modified during the archive attempt, ADSM does not archive the file.

### **Static**

Specifies that a file must not be modified during archive. If a file is modified during the archive attempt, it is not archived.

Platforms that do not support the STATIC option default to SHRSTATIC. The STATIC option is not supported on the DOS platform.

### **SHRDynamic**

Specifies that if the file is being modified during an archive attempt, ADSM archives the file during its last attempt even though the file is being modified. ADSM attempts to perform an archive as many as four times, depending on the value specified for CHANGINGRETRIES in the client user options file.

## DEFINE COPYGROUP

### Dynamic

Specifies that a file is archived on the first attempt by ADSM, regardless of whether the file is being modified during archive processing.

## Examples

### Task 1

Create a backup copy group named STANDARD for management class ACTIVEFILES in policy set VACATION in the EMPLOYEE\_RECORDS policy domain. The backup destination is BACKUPPOOL, with a minimum interval of three days between backups, regardless of whether the files have been modified. Retain up to five backup versions while the file resides on the node's file system.

**Command:** `define copygroup employee_records vacation activefiles standard  
type=backup destination=backuppool frequency=3 verexists=5  
mode=absolute`

### Output:

```
ANR1530I Backup copy group STANDARD defined in policy domain  
EMPLOYEE_RECORDS, set VACATION, management class ACTIVEFILES.
```

### Task 2

Create an archive copy group named STANDARD for management class MCLASS1 in policy set SUMMER in the PROG1 policy domain. The archive destination is ARCHIVEPOOL, where the archive copy is kept for 730 days.

**Command:** `define copygroup prog1 summer mclass1 standard type=archive  
destination=archivepool retver=730`

### Output:

```
ANR1535I Archive copy group STANDARD defined in policy domain  
PROG1, set SUMMER, management class MCLASS1.
```

## Related Commands

Table 33 provides references to related commands.

Table 33 (Page 1 of 2). Commands Related to DEFINE COPYGROUP

Command	Description	Reference
ASSIGN DEFMGMTCLASS	Assigns an existing management class as the default for a specified policy set.	Page 48

## DEFINE COPYGROUP

Table 33 (Page 2 of 2). Commands Related to DEFINE COPYGROUP

Command	Description	Reference
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DEFINE STGPOOL	Defines a storage pool as a named collection of data storage media.	Page 180
DELETE COPYGROUP	Deletes a backup or archive copy group from a policy domain and policy set.	Page 199
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539

## DEFINE DBBACKUPTRIGGER

---

### DEFINE DBBACKUPTRIGGER (Define the Database Backup Trigger)

Use the DEFINE DBBACKUPTRIGGER command to define settings for the database backup trigger. The database backup trigger determines when ADSM automatically runs a full or incremental backup of the ADSM database and deletes any unnecessary recovery log records.

ADSM uses the settings you specify with this command only when the log mode is set to ROLLFORWARD using the SET LOGMODE command.

With the DEFINE DBBACKUPTRIGGER command, you specify the percentage of the assigned capacity of the recovery log that can be used before ADSM begins a backup of the database. When space usage reaches the percentage you specify, ADSM starts a background process that does the following:

- Runs a backup of the ADSM database using one or more scratch volumes.

Unless a full backup is required, ADSM runs an incremental backup. A full backup is required when any of the following conditions are true:

- The database has never been backed up.
- The database was loaded using a salvage dump created with the DSMSEV DUMPDB command and the database has not been backed up since it was loaded.
- The maximum number of incremental backups allowed between full backups has been reached.
- The log mode was changed from NORMAL to ROLLFORWARD.
- The database was extended or reduced.

ADSM can continue to write new records to the recovery log while creating a backup of the database.

**Note:** If a backup initiated by the database backup trigger fails, ADSM waits 60 seconds, and then tries again. It continues to retry the backup until it is successful. If the mount limit for the device class being used for the backup has been reached (all drives busy), ADSM automatically cancels lower priority operations (such as reclamation) to make a mount point available for the database backup. To ensure that a mount point is always available to run a database backup, you can define a device class that is used only by database backup.

- After the backup is successfully completed, ADSM deletes all recovery log records except those required to restore the database to its most current state when used with the backup of the database that was just created.

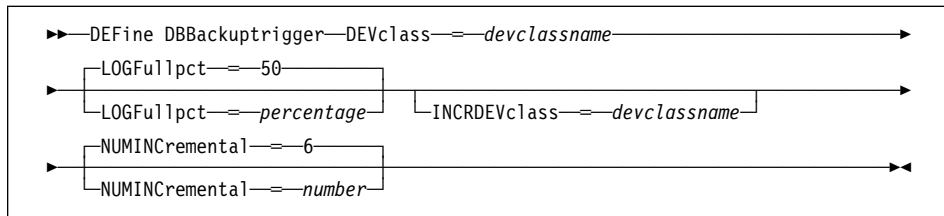
To display settings for the database backup trigger, use the QUERY DBBACKUPTRIGGER command. To update the settings for the database backup trigger, use the UPDATE DBBACKUPTRIGGER command. To delete the database backup trigger, use the DELETE DBBACKUPTRIGGER command.

## DEFINE DBBACKUPTRIGGER

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### **DEVclass=devclassname**

Specifies the name of the sequential access device class to use for backups. This parameter is required.

If the mount limit for the device class specified has already been reached (all drives are busy) when the backup runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the backup.

The server options file that is created during installation includes a DEVCONFIG option that specifies an external file named devcnfg.out. ADSM stores a backup copy of device class definitions in this file. If your ADSM database is damaged or lost and must be restored, the definitions created using the DEFINE DEVCLASS command will not be available and will need to be recreated manually.

The server options file also includes a VOLUMEHISTORY option that specifies an external file (named volhist.out) in which to store a backup copy of sequential volume history information. The volume history file is also used for database recovery.

#### **LOGFullpct=percentage**

Specifies when ADSM automatically backs up the database, based on the percentage of available space used for the recovery log. When the percentage of space used for the recovery log exceeds this value, ADSM performs an automatic backup of the database and removes unnecessary records from the recovery log. This parameter is optional. You can specify a value from 1–99. The default value is 50 percent.

#### **INCRDEVclass=devclassname**

Specifies the name of the sequential access device class to use for incremental backups. This parameter is optional. If you do not specify a value for this option, all backups are run using the device class specified with the DEVCLASS parameter.

## DEFINE DBBACKUPTRIGGER

If the mount limit for the device class specified has already been reached (all drives are busy) when the backup runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the backup.

The server options file that is created during installation includes a DEVCONFIG option that specifies an external file named devcnfg.out. ADSM stores a backup copy of device class definitions in this file. If your ADSM database is damaged or lost and must be restored, the definitions created using the DEFINE DEVCLASS command will not be available and will need to be recreated manually.

### **NUMIN**cremental=*number*

Specifies the maximum number of incremental backups that can be run before a full backup is required. This parameter is optional. You can specify a value from 0–32. A 0 specifies that ADSM runs only full backups automatically. The default value is 6.

Each incremental backup, whether run automatically by ADSM or run by issuing the BACKUP DB command, is added to the count of incremental backups run.

Each full backup, whether run automatically by ADSM or run by issuing the BACKUP DB command, resets the count for incremental backups to zero.

**Note:** If you issue a BACKUP DB command with the TYPE=INCREMENTAL parameter, ADSM performs an incremental backup of the database regardless of the NUMINCREMENTAL setting. For example, you set NUMINCREMENTAL to 5, and there have been five incremental backups since the last full backup. If you then issue BACKUP DB TYPE=INCREMENTAL, an incremental backup is still taken, and the counter for the number of incremental backups since the last full backup is set to 6. This occurs if the BACKUP DB command is issued either by an administrator or through an administrative schedule.

## Examples

### Task

Define the setting of the database backup trigger at 80 percent. Run two incremental backups to every full backup of the database. Assume a device class of FILE for the backup.

**Command:** DEFINE DBBackuptrigger DEVclass=file LOGFullpct=80 NUMINcremental=2

### Output:

```
ANR2283I Database backup trigger defined, but is disabled.
```

## DEFINE DBBACKUPTRIGGER

### Related Commands

Table 34 provides references to related commands.

*Table 34. Commands Related to DEFINE DBBACKUPTRIGGER*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DELETE DBBACKUPTRIGGER	Deletes the database backup trigger.	Page 202
QUERY DBBACKUPTRIGGER	Displays information about the database backup trigger.	Page 345
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514
UPDATE DBBACKUPTRIGGER	Changes one or more attributes of the database backup trigger.	Page 547

## DEFINE DBCOPY

---

### DEFINE DBCOPY (Define a Volume Copy of a Database Volume)

Use the DEFINE DBCOPY command to create a volume copy of an existing database volume. Database volume copies increase the availability of the database and provide safeguard against media failure. You can have as many as two volume copies of an existing database volume. After each database volume copy is defined, the server attempts to synchronize the volume copy with the database.

The volume copy created via the DEFINE DBCOPY command is also called a *mirrored volume*. The mirrorread and mirrorwrite options located in the server options file control the usage of mirrored volumes.

All database volumes must reside on a random access volume. Volume copies must have at least the same capacity as the original volume and should be defined on separate physical devices. Any additional space on the volume copy is not used.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If a DEFINE DBCOPY background process is canceled, the volume copy is created; however, synchronization processing is canceled, leaving the volume in a *stale* state. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
►►—DEFine DBCopy—volname—copyvolname—◄◄
```

### Parameters

#### *volname*

Specifies the name of the database volume for which you want to create a volume copy. This parameter is required.

#### *copyvolname*

Specifies the file name of the volume you want to assign as a database volume copy. This parameter is required. This file name can be entered with either the fully qualified path name or the path name relative to the server's current working directory. For example, if the file name is mainvol, the path name to be entered is:

```
d:\admserv\mainvol
```

No volume name used by ADSM can contain any embedded blanks or equal signs.



## DEFINE DBCOPY

### Examples

#### Task

Create a database volume copy named dvolcop for the database volume named dvol and located in d:\admserv\

**Command:** `define dbcopy d:\admserv\dvol d:\admserv\dvolcop`

#### Output:

```
ANR0220E Synchronization of database volume
d:\admserv\dvolcop started as process 204.
ANR2242I Database volume copy d:\admserv\
dvolcop defined; synchronization process
started (process ID 204).
ANR0234I Synchronization complete for database
volume d:\admserv\dvolcop.
```

### Related Commands

Table 35 provides references to related commands.

*Table 35. Commands Related to DEFINE DBCOPY*

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
DELETE DBVOLUME	Deletes a volume from the database.	Page 204
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## DEFINE DBVOLUME

---

### DEFINE DBVOLUME (Define a Database Volume)

Use the DEFINE DBVOLUME command to define a new database volume. All database volumes must reside on a random access volume.

**Attention:** The size of an allocated database, recovery log, or storage pool volume cannot be changed once it has been defined to the ADSM server. ADSM uses the initial size allocation of the volume at the time it is defined to the server to calculate data placement for later retrieval.

Each time you define a new database volume, you provide additional space to the database. To permit the database to use this additional space, use the EXTEND DB command.

When you define a database volume, it must occupy at least 5MB of space. Regardless of the defined storage capacity of a database volume, 1MB is not available for database storage. For example, if you define a database volume with a capacity of 9MB, 8MB are available for storage on the volume.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
►►—DEFine DBVolume—volname————◄◄
```

### Parameters

*volname*

Specifies the file name of the database volume to be defined. This parameter is required. This file name can be entered with either the fully qualified path name or the path name of the server's current working directory. For example, if the file name is mainvol, the path name to be entered is:

```
d:\adsmserve\mainvol
```

No volume name used by ADSM can contain any embedded blanks or equal signs.

## DEFINE DBVOLUME

### Examples

#### Task

Create a database volume in path d:\adsmserv\with a file name of dvol.

**Command:** `define dbvolume d:\adsmserv\dvol`

#### Output:

```
ANR2240I Database volume D:\ADSMSEV\DVOL defined.
```

### Related Commands

Table 36 provides references to related commands.

*Table 36. Commands Related to DEFINE DBVOLUME*

Command	Description	Reference
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DELETE DBVOLUME	Deletes a volume from the database.	Page 204
EXTEND DB	Extends usable database storage within the volumes previously allocated for its use.	Page 269
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347

## DEFINE DEVCLASS

---

### DEFINE DEVCLASS (Define a Device Class)

Use the DEFINE DEVCLASS command to define a device class for storage pools.

You can use the UPDATE DEVCLASS and DELETE DEVCLASS commands to either modify or eliminate existing device classes. However, you cannot modify the device type of an existing device class. The DISK device class is predefined by ADSM and requires no user intervention. This also means that the DISK device class cannot be modified or eliminated with the UPDATE DEVCLASS or DELETE DEVCLASS command.

To avoid confusion, it is beneficial to define a unique device class to support the sequential access devices available at your installation. For more information on defining device classes, refer to *ADSM Administrator's Guide*.

The server options file that is created during installation includes a DEVCONFIG option that specifies a default device configuration file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. The files you specify with this option are automatically updated with the results of this command. The files are also automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

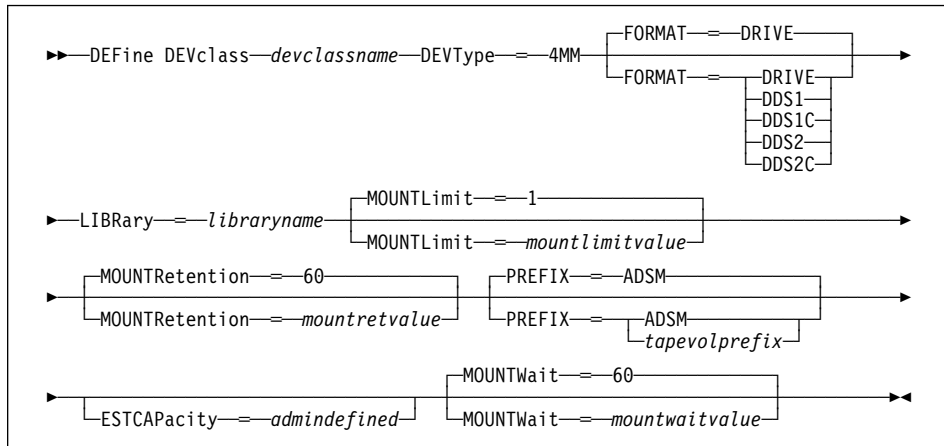
### Syntax

The supported parameters for this command vary between device types; therefore, the syntax and parameter descriptions are provided according to the device type. The syntax and parameter information is presented in the following order:

- 4MM
- 8MM
- DLT
- FILE
- LAN3995
- MO35
- OPTICAL
- QIC
- REMOVABLEFILE

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of 4MM



#### Parameters for Defining a Device Class with a Device Type of 4MM

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### DEVType=4MM

Specifies that the 4MM device type is assigned to the device class. This parameter is required. 4MM indicates that 4mm tape devices are assigned to this device class.

##### FORMAT=*tapeformat*

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE. Tape capacities are provided using their uncompressed values; hardware compression significantly increases these capacity values.

The following are possible values for 4mm devices:

##### DRIVE

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices is used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

##### DDS1

Specifies that ADSM writes data using the DDS-1 recording format for 4mm tapes, without compression. This format results in an uncompressed cartridge capacity of 1.3GB on 60 meter tape or 2GB on 90 meter tape. This format only applies to 60 and 90 meter tapes.

## DEFINE DEVCLASS

### **DDS1C**

Specifies that ADSM writes data using the DDS-1 recording format for 4mm tapes, with compression enabled. This results in a cartridge capacity of approximately 1.3GB on a 60 meter tape or 2GB on a 90 meter tape. This format, however, uses the tape drive hardware compression feature, so the actual capacity may be greater, depending on the effectiveness of compression. This format only applies to 60 and 90 meter tapes.

### **DDS2**

Specifies that ADSM writes data using the DDS-2 recording format for 4mm tapes, without compression. This format results in a cartridge capacity of 4GB on 120 meter tape. This format only applies to 120 meter tape.

### **DDS2C**

Specifies that ADSM writes data using the DDS-2 recording format for 4mm tapes, with compression. This format results in a cartridge capacity of approximately 8GB on 120 meter tape. This format, however, uses the tape drive hardware compression feature, so the actual capacity may be greater, depending on the effectiveness of compression. This format only applies to 120 meter tape.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the 4mm tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as, reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

## DEFINE DEVCLASS

If you have already established a media label naming convention that supports your current management system, use a volume prefix that conforms to your naming conventions.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for 4mm tapes, see Table 37.

*Table 37. ADSM Default Estimated Capacity for 4mm Tape*

<b>Tape Format</b>	<b>Estimated Capacity</b>
DDS1	1.3GB–2.0GB
DDS1C	1.3GB–2.0GB (see note)
DDS2	4.0GB
DDS2C	8.0GB (see note)

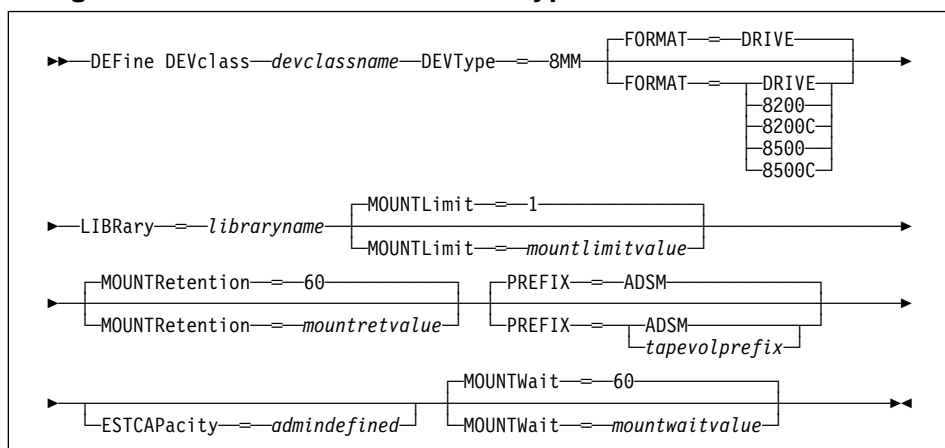
**Note:** This format uses the tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity may be greater than the listed value.

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of 8MM



#### Parameters for Defining a Device Class with a Device Type of 8MM

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### **DEVType=8MM**

Specifies that the 8MM device type is assigned to the device class. This parameter is required. 8MM indicates that 8mm tape devices are assigned to this device class.

##### **FORMAT=tapeformat**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

Possible values for 8mm devices:

##### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices is used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

##### **8200**

Specifies that ADSM writes data using the 8200 recording format. This results in a tape capacity of 2.3GB when using standard 112M tape cartridges.

##### **8200C**

Specifies that ADSM writes data using the 8200C recording format. This results in a tape capacity of approximately 2.3GB when using standard 112



## DEFINE DEVCLASS

meter tape cartridges. This format, however, uses the tape drive's hardware compression feature, so the actual capacity may be greater than 2.3GB, depending on the effectiveness of the compression feature.

### **8500**

Specifies that ADSM writes data using the 8500 recording format. This results in a tape capacity of 5.0GB when using standard 112M tape cartridges.

### **8500C**

Specifies that ADSM writes data using the 8500C recording format. This results in a tape capacity of at least 5.0GB when using standard 112M tape cartridges. However, this format uses the tape drive's hardware compression feature, so the actual capacity of the cartridge may be greater than 5.0GB, depending on the effectiveness of the compression feature.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the 8mm tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as, reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

If you have already established a media label naming convention that supports your current management system, use a volume prefix that conforms to your naming conventions.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter

## DEFINE DEVCLASS

if the default estimated capacity for the device class is inaccurate due to compression of data.

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for 8mm tapes, see Table 38.

Table 38. ADSM Default Estimated Capacity for 8mm Tape

Tape Format	Estimated Capacity
8200	2.3GB
8200C	2.3GB (see note)
8500	5.0GB
8500C	5.0GB (see note)

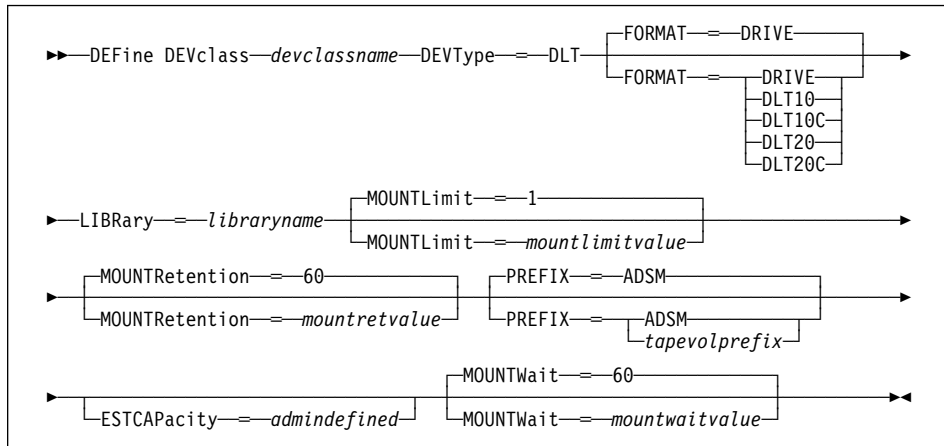
**Note:** This format uses the tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity may be greater than the listed value.

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of DLT



#### Parameters for Defining a Device Class with a Device Type of DLT

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### DEVType=DLT

Specifies that the DLT device type is assigned to the device class. This parameter is required. DLT indicates that DLT devices are assigned to this device class.

##### FORMAT=*tapeformat*

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

These are the possible values for DLT devices:

##### DRIVE

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices is used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

##### DLT10

Specifies that ADSM writes data using the DLT10 recording format. This results in a tape capacity of approximately 10.0GB when using CompactTape III or CompactTape IV cartridges.

##### DLT10C

Specifies that ADSM writes data using the DLT10C recording format. This results in a tape capacity of approximately 10.0GB when using

## DEFINE DEVCLASS

CompacTape III and CompacTape IV cartridges. This format, however, uses the tape drive's hardware compression feature, so the actual capacity may be greater than 10.0GB, depending on the effectiveness of the compression feature.

### **DLT20**

Specifies that ADSM writes data using the DLT20 recording format. This results in a tape capacity of approximately 20.0GB when using CompacTape IV cartridges.

**Attention:** The DLT20 format can only be used with CompacTape IV cartridges in a DLT4000 drive.

### **DLT20C**

Specifies that ADSM writes data using the DLT20C recording format. This results in a tape capacity of approximately 20.0GB when using CompacTape IV cartridges. This format, however, uses the tape drive's hardware compression feature, so the actual capacity of the cartridge may be greater than 20.0GB, depending on the effectiveness of the compression feature.

**Attention:** The DLT20C format can only be used with CompacTape IV cartridges in a DLT4000 drive.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the DLT drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as, reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

## DEFINE DEVCLASS

If you have already established a media label naming convention that supports your current management system, use a volume prefix that conforms to your naming conventions.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for DLT, see Table 39.

*Table 39. ADSM Default Estimated Capacity for DLT*

<b>Tape Format</b>	<b>Estimated Capacity</b>
DLT10	10.0GB
DLT10C	10.0GB (see note)
DLT20	20.0GB
DLT20C	20.0GB (see note)

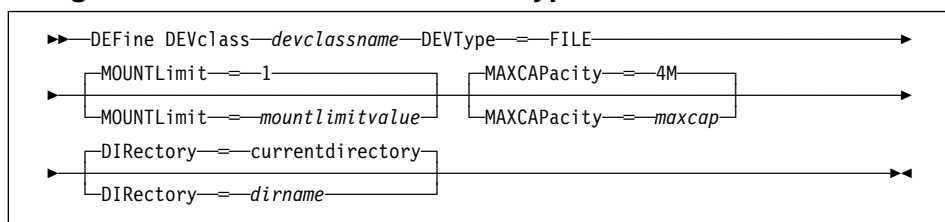
**Note:** This format uses the tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity may be greater than the listed value.

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of FILE



#### Parameters for Defining a Device Class with a Device Type of FILE

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### **DEVType=FILE**

Specifies that the FILE device type is assigned to the device class. This parameter is required. FILE indicates that a file or files are assigned to this device class. When the server needs to access a volume that belongs to this device class, it opens a file and reads or writes file data. A file is a form of sequential access media.

##### **MOUNTLimit=mountlimitvalue**

Specifies the maximum number of files that can be simultaneously open for input/output. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive.

##### **MAXCAPacity=maxcap**

Specifies the maximum size of any data storage files defined to a storage pool categorized by this device class. This parameter is optional. You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, MAXCAPACITY=5M specifies that the maximum capacity for a volume in this device class is 5 megabytes. The default value is 4MB (that is, MAXCAPACITY=4M). The smallest value allowed is 100KB (that is, MAXCAPACITY=100K).

##### **DIRectory=dirname**

Specifies the directory location of the file or files used in this device class. This parameter is optional. The default is the current working directory of the server at the time the command is issued, unless the DSMSEV\_DIR environment variable has been set. For additional information about the DSMSEV\_DIR variable, refer to *ADSM Quick Start*.

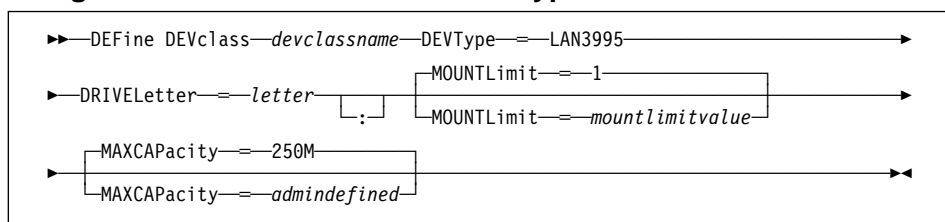
## DEFINE DEVCLASS

By specifying a directory name, you identify the location where the server places the files that represent storage volumes for this device class. While processing the command, the server expands the specified directory name into its fully qualified form, starting from the root directory.

Later, if the server needs to allocate a scratch volume, it creates a new file in this directory. For scratch volumes used to store client data, the file created by the server has a file name extension of .BFS. For scratch volumes used to store export data, a file name extension of .EXP is used. For example, suppose you define a device class with a DIRECTORY of **\\ADSMSTOR** and the server needs a scratch volume in this device class to store export data. The file which the server creates might then be named **\\ADSMSTOR\00566497.EXP**.

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of LAN3995



#### Parameters for Defining a Device Class with a Device Type of LAN3995

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### **DEVType=LAN3995**

Specifies the LAN3995 device type is assigned to the device class. This parameter is required. LAN3995 indicates that a LAN-attached 3995 optical library is assigned to this device class.

##### **DRIVELetter=letter**

Specifies the letter that identifies the drive. This parameter is required; however, the drive letter colon is optional.

##### **MOUNTLimit=mountlimitvalue**

Specifies the maximum number of volumes that can be simultaneously open for input/output. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class in the 3995 library.

##### **MAXCAPacity=maxcap**

Specifies the maximum amount of data to store on each side of a volume in a storage pool associated with this device class. Since each volume has two sides, the maximum capacity of a volume is twice this value. This parameter is optional. The default is 250M.

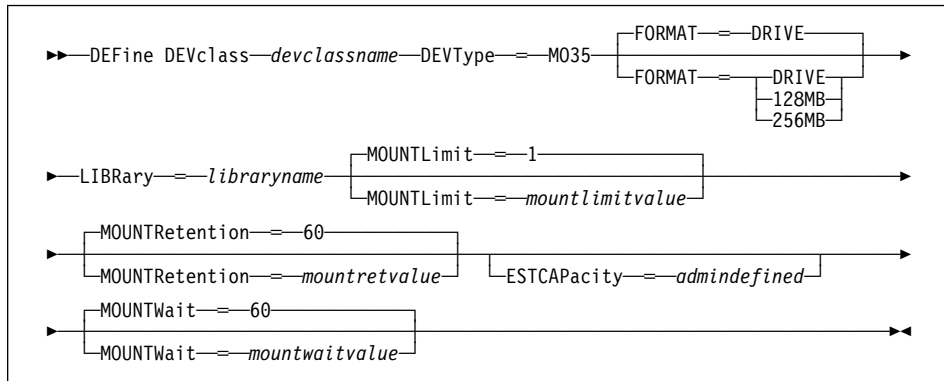
This value is specified as an integer followed by a scale factor. You must specify a scale factor following the specified integer. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte



## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of MO35



#### Parameters for Defining a Device Class with a Device Type of MO35

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### DEVType=MO35

Specifies the MO35 device type is assigned to the device class. This parameter is required. MO35 indicates that three and one-half inch optical disk devices are assigned to this device class.

##### FORMAT=*optformat*

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

There are two possible values for optical devices:

##### DRIVE

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Use the DRIVE value only when the device class has compatible sequential access devices. If you specify DRIVE for a device class that has incompatible sequential access devices, then you must mount volumes on devices that are capable of reading or writing at the format established when the volume is first mounted. This can cause errors if the only sequential access device that can access the volume is already in use.

##### 128MB

Specifies that ADSM writes data to a 128MB three and one-half inch optical drive.

##### 256MB

Specifies that ADSM writes data to a 256MB (230MB) three and one-half inch optical drive.

## DEFINE DEVCLASS

**LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the optical drives which can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

**MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class.

If this value is set to 1, then it may not be possible for media-to-media operations (like reclamation) to work.

**MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

## DEFINE DEVCLASS

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes defined to a storage pool categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

This value is specified as an integer followed by a scale factor. You must specify a scale factor following the specified integer. For example, ESTCAPACITY=5M specifies that the estimated capacity for this device class is 5 megabytes. The following are possible scale factors:

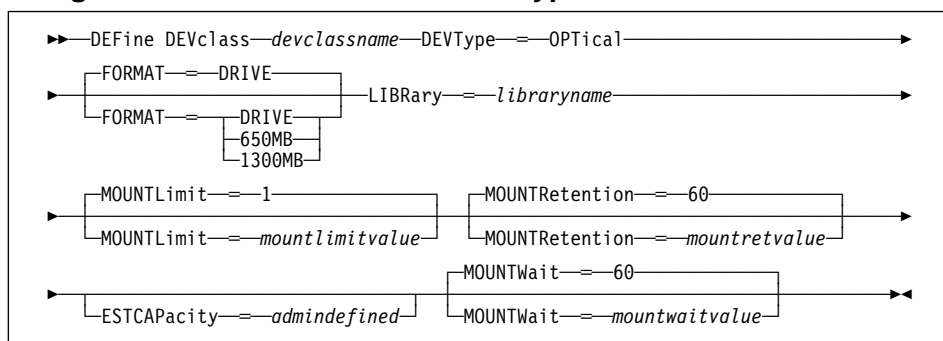
<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of OPTICAL



### Parameters for Defining a Device Class with a Device Type of OPTICAL

#### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

#### **DEVType**=*optical*

Specifies that the OPTICAL device type is assigned to the device class. This parameter is required. OPTICAL indicates that the device class uses two-sided 5.25 inch rewritable optical media.

#### **FORMAT**=*optformat*

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

Possible values are:

#### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices is used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

#### **650MB**

Specifies that ADSM writes data to a 650 MB 5.25-inch optical drive.

#### **1300MB**

Specifies that ADSM writes data to a 1300 MB 5.25-inch optical drive.

#### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the optical drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

## DEFINE DEVCLASS

**MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as, reclamation) to work.

**MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

**ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

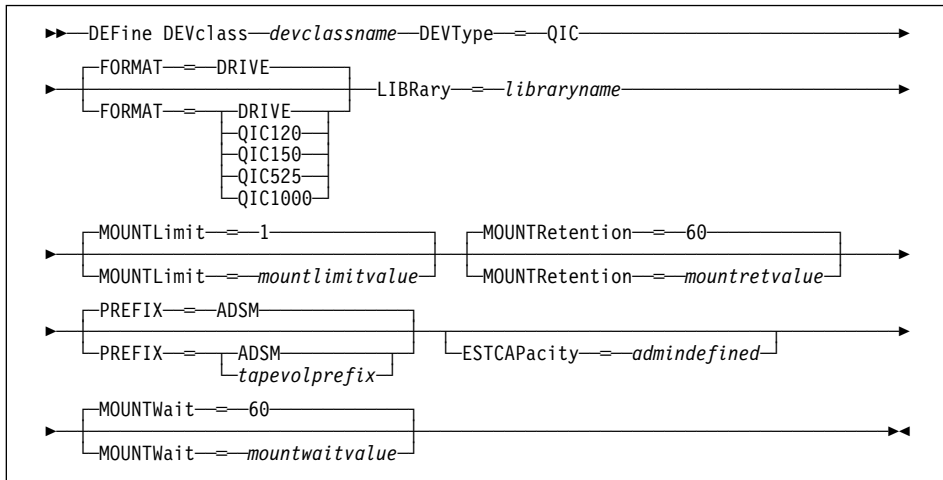
For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

**MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of QIC



### Parameters for Defining a Device Class with a Device Type of QIC

#### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

#### DEVType=QIC

Specifies that the QIC device type is assigned to the device class. This parameter is required. QIC indicates that quarter-inch cartridge tape devices are assigned to this device class.

#### FORMAT=*tapeformat*

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

The following are possible values for QIC devices:

**DRIVE** Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices is used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

**QIC120** See Table 40 on page 139.

**QIC150** See Table 40 on page 139.

**QIC525** See Table 40 on page 139.

**QIC1000** See Table 40 on page 139.

## DEFINE DEVCLASS

Table 40. QIC Tape Recording Format Selections

Tape/Format	QIC-120	QIC-150	QIC-525	QIC-1000
3M DC300XLP	–	–	–	–
3M DC600A	Read	–	–	–
3M DC600XTD	Read/Write	Read/Write	–	–
3M DC6150	Read/Write	Read/Write	–	–
3M DC6320	Read/Write	Read/Write	Read/Write	–
3M DC6525	Read/Write	Read/Write	Read/Write	–
3M DC9100	–	–	–	Read/Write
3M DC9120XL	–	–	–	Read/Write

**Note:** ADSM cannot use 3M DC300XLP and 3M DC600A tapes.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the QIC tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as, reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

If you have already established a media label naming convention that supports your current management system, use a volume prefix that conforms to your naming conventions.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

## DEFINE DEVCLASS

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for QIC tapes, see Table 41.

Table 41. ADSM Default Estimated Capacity for QIC Tape

Tape Format	Estimated Capacity (range)
QIC120	26MB–172MB
QIC150	31MB–207MB
QIC525	65MB–427MB
QIC1000	169MB–1.1GB

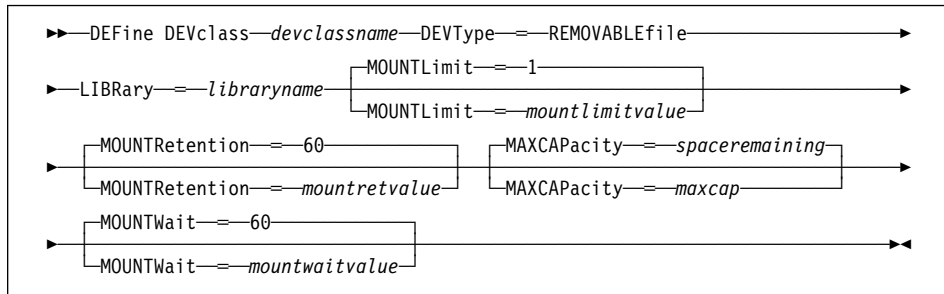
### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.



## DEFINE DEVCLASS

### Syntax for Defining a Device Class with a Device Type of REMOVABLEFILE



#### Parameters for Defining a Device Class with a Device Type of REMOVABLEFILE

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### **DEVType=REMOVABLEfile**

Specifies that the REMOVABLEFILE device type is assigned to the device class. This parameter is required. REMOVABLEFILE indicates that the volumes for this device class are files on local, removable media. To access volumes that belong to this device class, the server requests that the removable media be mounted in drives. The server then opens a file on the media and reads or writes file data. Common drives for this device class include Iomega Zip and Jaz drives and SyQuest drives. Removable file drives are treated as single sided. Two-sided media are treated as two separate volumes.

The server recognizes that the media can be removed and that additional media can be inserted, subject to limits set with the MOUNTLIMIT parameter for the device class and the MAXSCRATCH parameter for the storage pool.

Volumes in a device class with device type REMOVABLEFILE are sequential access volumes.

##### **LIBRARY=libraryname**

Specifies the name of the defined library object that contains the removable media drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

##### **MOUNTLimit=mountlimitvalue**

Specifies the maximum number of files that can be simultaneously open for input/output in this device class. This parameter is optional. The default value is 1. You can specify an integer in the range of 1 to 256, inclusive.

Each removable medium contains only one file. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as reclamation) to work.

## DEFINE DEVCLASS

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **MAXCAPacity**=*maxcap*

Specifies the maximum size of any volumes (which are actually files) defined to a storage pool categorized by this device class. This parameter is optional. You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, MAXCAPACITY=5M specifies that the maximum capacity for a volume in this device class is 5 megabytes. The default max capacity is the space remaining on the media after it is first used. The smallest value allowed is 100KB (that is, MAXCAPACITY=100K).

Because the server opens only one file per physical removable medium, specify a capacity such that the one file makes full use of your media capacity.

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. The default value is 60 minutes. You can specify an integer in the range of 0 to 9999, inclusive.

## Examples

### Task 1

Define a device class named PLAINFILES with a file device type and a maximum capacity of 50MB.

**Command:** `define devclass plainfiles devtype=file maxcapacity=50m`

## DEFINE DEVCLASS

### Output:

```
ANR2203I Device class PLAINFILES defined.
```

### Task 2

Define a device class named 8MMTAPE for an 8mm device in a library named AUTO. The format is DRIVE, mount limit is 2, mount retention is 10, tape volume prefix is named ADSMVOL, and the estimated capacity is 6GB.

**Command:** `define devclass 8mmtape devtype=8mm library=auto format=drive mountlimit=2 mountretention=10 prefix=adsmvol estcapacity=6g`

### Output:

```
ANR2203I Device class 8MMTAPE defined.
```

## Related Commands

Table 42 provides references to related commands.

Command	Description	Reference
BACKUP DEVCONFIG	Backs up ADSM device information to a file.	Page 66
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE DEVCLASS	Deletes a device class name.	Page 206
QUERY DEVCLASS	Displays information about one or more device classes.	Page 350
UPDATE DEVCLASS	Changes the attributes of a device class.	Page 550

## DEFINE DOMAIN

---

### DEFINE DOMAIN (Define a New Policy Domain)

Use the DEFINE DOMAIN command to define a new policy domain.

A policy domain provides you with a logical way of managing backup and archive policies for a group of client nodes with common backup-archive policy needs. You can also define policy domains to limit the number of client nodes that can be managed by a single policy administrator. A policy domain contains one or more policy sets, with only one policy set active at a time.

Each policy set contains one or more management classes. You can specify one default management class for each policy set. A management class can contain a backup copy group, an archive copy group, or both; however, neither is required.

An ADSM client node is assigned to one policy domain. The active policy set within the policy domain should contain management classes that meet the needs of files stored on the client node. The user of the client node can use the default management class or select any management class in the active policy set to manage files that are backed up or archived from the client node. Client nodes running on different platforms can be assigned to the same policy domain. For example, a policy domain can be used by both OS/2 and AIX clients.

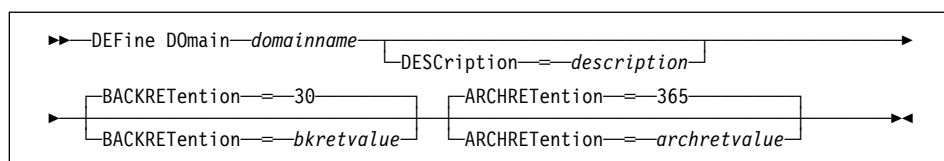
This command specifies the backup and archive retention grace periods for the policy domain. These retention grace periods are used by ADSM to protect backup versions and archive copies from deletion when the respective copy group is not available. For further descriptions of these parameters, see the parameter definitions. You can also find a detailed discussion of the retention grace periods in the *ADSM Administrator's Guide*. To determine the retention periods for an existing policy domain, use the QUERY DOMAIN command.

After you define a new policy domain, you can define additional policy objects in the new policy domain, such as policy sets, management classes, and copy groups. You can use COPY commands to copy policy sets and management classes. In addition, you must activate a policy set before clients assigned to the new policy domain can back up or archive files. A new policy domain can also be created by copying an existing policy domain to a new policy domain name.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



## DEFINE DOMAIN

### Parameters

*domainname*

Specifies the name of the policy domain to be defined. This parameter is required. The maximum length of this name is 30 characters.

**DESCRiption**=*description*

Specifies a text string that describes the policy domain. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.

**BACKRETention**=*bkretvalue*

Specifies an integer that defines the number of days (from the date of deactivation) to retain backup versions that are no longer on the client's system. This parameter is optional. The default value is 30. You can specify an integer in the range of 0 to 9999, inclusive. The backup retention value is used by ADSM to manage inactive versions of files when any of the following conditions are present:

- A file's management class is changed, but neither the new management class nor the default management class contains a backup copy group.
- The management class to which a file is bound no longer exists, and the default management class does not contain a backup copy group.
- The backup copy group is deleted from the management class to which a file is bound and the default management class does not contain a backup copy group.

**ARCHRETention**=*archretvalue*

Specifies an integer that defines the number of days (from the date of archive) to retain archive copies. This parameter is optional. The default value is 365. You can specify an integer in the range of 0 to 9999, inclusive. The archive retention value is used by ADSM to manage inactive copies of files when either of the following conditions are present:

- The management class to which a file is bound no longer exists, and the default management class does not contain an archive copy group.
- The archive copy group is deleted from the management class to which a file is bound and the default management class does not contain an archive copy group.

### Examples

#### Task 1

Define a policy domain with the name of EMPLOYEE\_RECORDS.

## DEFINE DOMAIN

**Command:** DEFine D0main employee\_records

**Output:**

```
ANR1500I Policy domain EMPLOYEE_RECORDS defined.
```

### Task 2

Define a policy domain with a name of PROG1 and a description of Programming Group Domain. Specify that archive copy retention is for 90 days when management classes or archive copy groups are deleted and the default management class does not contain an archive copy group. Also specify that backup version retention is for 60 days when management classes or copy groups are deleted and the default management class does not contain a backup copy group.

**Command:** DEFine D0main prog1 DESCription="Programming Group Domain"  
BACKRETention=60 ARCHRETention=90

**Output:**

```
ANR1500I Policy domain PROG1 defined.
```

## Related Commands

Table 43 provides references to related commands.

Table 43. Commands Related to DEFINE DOMAIN

Command	Description	Reference
COPY DOMAIN	Creates a copy of an existing policy domain.	Page 89
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE DOMAIN	Deletes a policy domain along with any policy objects in the policy domain.	Page 208
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
UPDATE DOMAIN	Changes the attributes of a policy domain.	Page 573

## DEFINE DRIVE

---

### DEFINE DRIVE (Define a Drive to a Library)

Use the DEFINE DRIVE command to define a drive that ADSM can use to access data on storage volumes. Each drive is assigned to a single library, which must be defined prior to issuing this command. You can define more than one drive for a given library; this is done by issuing the DEFINE DRIVE command once for each drive.

**Remember:** Stand-alone drives always require a library.

For additional information, see *ADSM Administrator's Guide*.

### Privilege Class

To issue this command you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶—DEFine DRive—libname—drivename—DEVIce—devicename—▶
▶ [ELEMent—elementaddress—(1)]▶
```

**Note:**  
1 The ELEMENT parameter is required for drives being defined in SCSI libraries that support more than one drive and is optional for SCSI libraries that support only one drive. For additional information, see *ADSM Administrator's Guide*.

### Parameters

#### *libname*

Specifies the name of the library to which the drive is to be assigned. This parameter is required for all drives, including stand-alone drives. It is also required for drives that are device type REMOVABLEFILE. The specified library must have been previously defined using the DEFINE LIBRARY command.

#### *drivename*

Specifies the name assigned to the drive. This parameter is required. The maximum length of this name is 30 characters.

#### DEVIce=*devicename*

Specifies the name of the device that corresponds to the drive. For example, you might specify TAPE05\$. For a drive that is device type REMOVABLEFILE, OPTICAL, or M035, specify its drive letter and a colon, for example, E:. This parameter is required.

#### ELEMent=*elementaddress*

Specifies the element address of the drive. The ELEMENT parameter is required for drives being defined in SCSI libraries that support more than one drive; it is optional for SCSI libraries that support only one drive. The element address is a number that indicates the physical location of a drive within an automated library. ADSM needs the element address to connect the physical location of the drive to

## DEFINE DRIVE

the drive's SCSI address. For a description of which element address values to use for various library configurations, see *ADSM Administrator's Guide*.

### Usage Notes

- When issuing the DEFINE DRIVE command for a removable drive, a formatted medium must be present in the drive. For a REMOVABLEFILE device type, ADSM does not supply utilities to format or label the media. You must use the device manufacturer's utilities or operating system utilities to format and label the media. Such utilities may include the format and label commands. The label on the media must not contain more than 8 characters, embedded blanks, or periods. The label must also follow the operating system's rules for file names. This allows ADSM to verify that a valid file system is available.
- If your SCSI adapter has the capability of letting the system see attached drives as either fixed disks or removable, make sure the system views the attached drive as removable.

### Examples

#### Task 1

Suppose you have already defined a manual tape library named MANLIB using the DEFINE LIBRARY command. You then wish to inform the server that you have a tape drive that can be used to access storage media. You choose the name TAPEDRIVE0 for this drive.

**Command:** `define drive manlib tapedrive0 device=tape05$`

**Output:**

```
ANR8404I Drive TAPEDRIVE0 defined in library MANLIB.
```

#### Task 2

Suppose you have already defined an optical library named OPTICA using the DEFINE LIBRARY command. You then wish to inform the server that you have a stand-alone five and a quarter inch optical drive, accessed as drive letter Q, ready for use. You choose the name OPTDRIVE2 for this drive.

**Command:** `define drive optica optdrive2 device=q:`

**Output:**

```
ANR8404I Drive OPTDRIVE2 defined in library OPTICA.
```



## DEFINE DRIVE

### Task 3

Suppose you have already defined a manual library named MANUAL using the DEFINE LIBRARY command. You then wish to inform the server that you have a stand-alone Iomega Zip drive accessed as drive letter H, ready for use. You choose the name ZIP1 for this drive.

**Command:** `define drive manual ZIP1 device=h:`

**Output:**

```
ANR8404I Drive ZIP1 defined in library MANUAL.
```

### Related Commands

Table 44 provides references to related commands.

*Table 44. Commands Related to DEFINE DRIVE*

Command	Description	Reference
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE LIBRARY	Deletes a library.	Page 220
QUERY DRIVE	Displays information about one or more drives.	Page 359
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
UPDATE DRIVE	Changes the attributes of a drive.	Page 575

## DEFINE LIBRARY

---

### DEFINE LIBRARY (Define a Library)

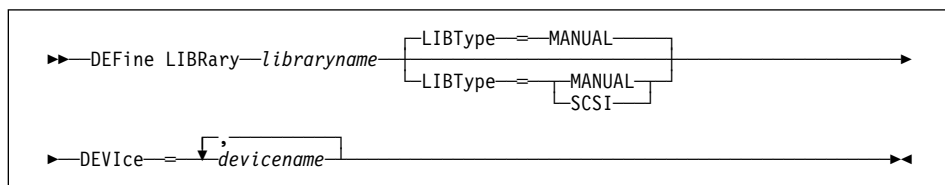
Use the DEFINE LIBRARY command to define a library and specify its type. A library is a collection of one or more drives, and possibly robotic devices (depending on the library type), which can be used to access storage volumes.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### *libraryname*

Specifies the name of the library to be defined. This parameter is required. The maximum length of this name is 30 characters.

#### **LIBType=librarytype**

Specifies the type of library that is being defined. This parameter is optional. The default is MANUAL. Possible values are:

#### **MANUAL**

Specifies that the library type is not automated and requires human interaction (a mount operator) when a storage volume needs to be mounted into a drive. When volumes need to be mounted on drives that reside in this type of library, messages will be sent to mount operators to inform them of the required action. This type of library is used with stand-alone drives.

#### **SCSI**

Specifies that the library is automated via a SCSI autochanger device. When volumes need to be mounted on drives that reside in this type of library, ADSM uses the robotic autochanger device to automatically carry out the required actions.

#### **DEVIce=devicename**

Specifies the names of the devices that access the library's robotic mechanism. For example, you might specify auto04\$ as the device name. This parameter is

## DEFINE LIBRARY

required for SCSI, but is not valid for MANUAL library types. If the library type is SCSI, only one device name is allowed. For additional information, see the *ADSM Administrator's Guide*.

### Examples

#### Task 1

Define a library named MANLIB with the library type of MANUAL.

**Command:** `DEFine LIBRary manlib LIBType=MANUAL`

**Output:**

```
ANR8400I Library MANLIB defined.
```

#### Task 2

Define a library named EZLIFE with a library type of SCSI and a device name of AUTO04\$.

**Command:** `DEFine LIBRary ezlife LIBType=SCSI DEvIce=auto04$`

**Output:**

```
ANR8400I Library EZLIFE defined.
```

### Related Commands

Table 45 provides references to related commands.

*Table 45 (Page 1 of 2). Commands Related to DEFINE LIBRARY*

Command	Description	Reference
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DELETE LIBRARY	Deletes a library.	Page 220
QUERY DRIVE	Displays information about one or more drives.	Page 359
QUERY LIBRARY	Displays information about one or more libraries.	Page 381

## DEFINE LIBRARY

*Table 45 (Page 2 of 2). Commands Related to DEFINE LIBRARY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
UPDATE DRIVE	Changes the attributes of a drive.	Page 575
UPDATE LIBRARY	Changes the attributes of a library.	Page 577

## DEFINE LOGCOPY

---

### DEFINE LOGCOPY (Define a Volume Copy of a Recovery Log Volume)

Use the DEFINE LOGCOPY command to create a volume copy of an existing recovery log volume. Recovery log volume copies increase the availability of the recovery log and provide a safeguard against media failure. You can have as many as two volume copies of an existing recovery log volume. After each recovery log volume copy is defined, the server attempts to synchronize the volume copy with the recovery log.

The volume copy created via the DEFINE LOGCOPY command is also called a *mirrored volume*. The mirrorread and mirrorwrite options located in the server options file control the usage of mirrored volumes.

All recovery log volumes must reside on a random access volume. Volume copies must have at least the same capacity as the original volume and should be defined on separate physical devices. Any additional space on the volume copy is not used.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If a DEFINE LOGCOPY background process is canceled, the volume copy is created; however, synchronization processing is canceled leaving the volume in a *stale* state. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
►►—DEFine LOGCopy—volname—copyvolname—————◄◄
```

### Parameters

#### *volname*

Specifies the name of the recovery log volume for which you want to create a volume copy. This parameter is required.

#### *copyvolname*

Specifies the file name of the volume you want to assign as a recovery log volume copy. This parameter is required. This file name can be entered with either the fully qualified path name or the path name relative to the server's current working directory. For example, if the file name is mainvol, the path name to be entered is:

```
c:\admserv\mainvol
```

No volume name used by ADSM can contain any embedded blanks or equal signs.

## DEFINE LOGCOPY

### Examples

#### Task

Create a volume copy named log.1b of the recovery log volume named log.1a located in c:\adsmserve.

**Command:** define logcopy c:\adsmserve\log.1a c:\adsmserve\log.1b

#### Output:

```
ANR2261I Recovery log volume copy c:\adsmserve\log.1b defined;  
synchronization process started (process ID 88).
```

### Related Commands

Table 46 provides references to related commands.

Table 46. Commands Related to DEFINE LOGCOPY

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
DELETE LOGVOLUME	Deletes a volume from the recovery log.	Page 222
QUERY LOGVOLUME	Displays information about one or more log volumes.	Page 394
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## DEFINE LOGVOLUME

---

### DEFINE LOGVOLUME (Define a Recovery Log Volume)

Use the DEFINE LOGVOLUME command to define a new recovery log volume. All recovery log volumes must reside on a random access volume.

**Attention:** The size of an allocated database, recovery log, or storage pool volume cannot be changed once it has been defined to the ADSM server. ADSM uses the initial size allocation of the volume at the time it is defined to the server to calculate data placement for later retrieval.

Each time you define a new recovery log volume, you provide additional space for the recovery log. To permit the recovery log to use this additional space, use the EXTEND LOG command.

When you define a recovery log volume, it must occupy at least 5MB of space. Storage space is used in 4MB blocks after 1MB is used for overhead purposes. Regardless of the defined storage capacity of a recovery log volume, 1MB is not available for recovery log storage. For example, if you define a recovery log volume with a capacity of 9MB, 8MB are available for storage on the volume.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
►—DEFine LOGVolum—volname—◄◄
```

### Parameters

*volname*

Specifies the file name of the recovery log volume. This parameter is required. This file name can be entered with either the fully qualified path name or the path name of the server's current working directory. For example, if the file name is mainvol, the path name to be entered is:

```
d:\admserv\mainvol
```

No volume name used by ADSM can contain any embedded blanks or equal signs.

### Examples

#### Task

Define a recovery log volume in d:\admserv\ with a file name of logv.

## DEFINE LOGVOLUME

**Command:** define logvolume d:\admserv\logv

**Output:**

```
ANR2260I Recovery log volume D:\ADMSERV\LOGV defined.
```

### Related Commands

Table 47 provides references to related commands.

*Table 47. Commands Related to DEFINE LOGVOLUME*

Command	Description	Reference
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DELETE LOGVOLUME	Deletes a volume from the recovery log.	Page 222
EXTEND LOG	Extends usable log storage within the volumes previously allocated for its use.	Page 271
QUERY LOG	Displays allocation information about the recovery log.	Page 389
QUERY LOGVOLUME	Displays information about one or more log volumes.	Page 394



## DEFINE MGMTCLASS

### DEFINE MGMTCLASS (Define a Management Class)

Use the DEFINE MGMTCLASS command to define a new management class in a policy set.

You can define one or more management classes for each policy set in a policy domain. A management class can contain a backup copy group, an archive copy group, or both. However, neither is required.

**Note:** The DEFINE MGMTCLASS command will fail if a copy storage pool is specified as the destination for space-managed files.

You must specify an existing management class as the default management class for a policy set before you can activate the policy set. To specify an existing management class as the default management class, use the ASSIGN DEFMGMTCLASS command.

When a client node is registered with ADSM, the node is defined to a policy domain. This policy domain contains management classes appropriate for the information stored on the client node.

The user of a client node can use the default management class or select any management class in the active policy set. By entering clear descriptions of your management classes, users can easily select management classes that are appropriate for their storage management needs.

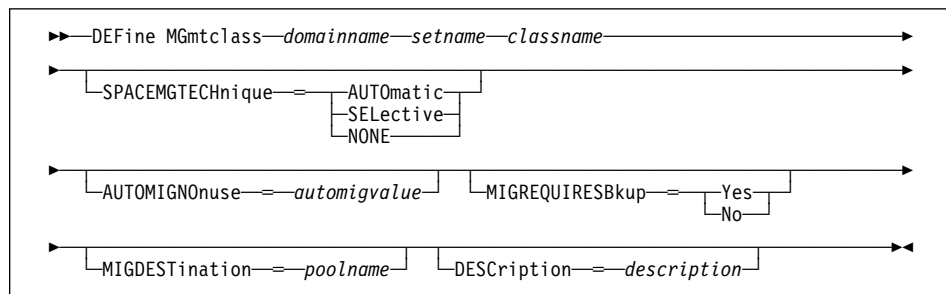
The following parameters affect only HSM clients that use this management class, and have no effect on backup-archive clients:

SPACEMGTECHNIQUE  
AUTOMIGNONUSE  
MIGREQUIRESBKUP  
MIGDESTINATION

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the management class belongs.

### Syntax



## DEFINE MGMTCLASS

### Parameters

#### *domainname*

Specifies the name of the policy domain to which the management class belongs. This parameter is required.

#### *setname*

Specifies the name of the policy set to which the management class belongs. This parameter is required. You cannot define a management class to the ACTIVE policy set.

#### *classname*

Specifies the name of the new management class. This parameter is required. The maximum length of this name is 30 characters. You cannot use either *default* or *grace\_period* as a class name.

#### **SPACEMGTECHnique=*smtvalue***

Specifies whether the file is eligible for migration. This parameter is optional. Possible values are:

##### **AUTOMATIC**

Specifies that the file is eligible for automatic migration or selective migration.

##### **SELECTIVE**

Specifies that the file is eligible for selective migration only.

##### **NONE**

Specifies that the file is not eligible for migration.

The default in the management class named STANDARD is NONE.

This parameter is effective only for HSM clients, not for backup-archive clients.

#### **AUTOMIGNOnuse=*automigvalue***

Specifies the number of days that must elapse since the file was last accessed before it is eligible for automatic migration. This parameter is optional. If SPACEMGTECHNIQUE is not AUTOMATIC, ADSM ignores this attribute. You can specify a value from 0–9999. The default value is 0.

This parameter is effective only for HSM clients, not for backup-archive clients.

#### **MIGREQUIRESBkup=*migreqvalue***

Specifies whether a backup version of a file must exist before a file can be migrated. This parameter is optional. Possible values are:

**Yes** A backup version must exist.

**No** A backup version is optional.

**Note:** This parameter is effective only for HSM clients, not for backup-archive clients. For HSM clients, the default is YES.

## DEFINE MGMTCLASS

### **MIGDESTination**=*poolname*

Specifies the name of a primary storage pool that is a destination for migrated files. A file must be associated with a management class that specifies an appropriate storage pool for migrated files. The type of storage devices in a storage pool and options specified for the storage pool affects the speed with which a migrated file can be recalled to a local file system. This parameter is optional. The default value is SPACEMGPOOL.

This parameter is effective only for HSM clients, not for backup-archive clients.

### **DESCRiption**=*description*

Describes the management class by using a text string. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters.

## Examples

### Task 1

Define a management class called ACTIVEFILES for policy set VACATION in the EMPLOYEE\_RECORDS policy domain.

**Command:** `define mgmtclass employee_records vacation activefiles`

**Output:**

```
ANR1520I Management class ACTIVEFILES defined in policy
domain EMPLOYEE_RECORDS, set VACATION.
```

### Task 2

Define a management class called MCLASS1, described as Technical Support Mgmt Class, for policy set SUMMER in the PROG1 policy domain.

**Command:** `define mgmtclass prog1 summer mclass1 description="technical support mgmt class"`

**Output:**

```
ANR1520I Management class MCLASS1 defined in policy domain
PROG1, set SUMMER.
```

## DEFINE MGMTCLASS

### Related Commands

Table 48 provides references to related commands.

*Table 48. Commands Related to DEFINE MGMTCLASS*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ASSIGN DEFMGMTCLASS	Assigns an existing management class as the default for a specified policy set.	Page 48
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581

## DEFINE POLICYSET

---

### DEFINE POLICYSET (Define a Policy Set)

Use the DEFINE POLICYSET command to define a policy set in a specified policy domain.

A policy set provides you with a logical way of grouping management classes and a set of definitions for use in the ACTIVE policy set. You can define one or more policy sets for each policy domain; however, only one policy set can be active at a time in a given policy domain.

A policy set can contain zero or more management classes. You can specify one default management class for each policy set. A management class can contain a backup copy group, an archive copy group, or both. However, neither is required.

Use the ACTIVATE POLICYSET command to make a new policy set the active policy set. When a policy set is activated, ADSM uses these new definitions to manage client files for the client nodes assigned to the policy domain where this newly activated policy set resides. Use the VALIDATE POLICYSET command to verify that a policy set is complete and valid before activating it with the ACTIVATE POLICYSET command.

After you define a policy set, you can define policy objects to the new policy set, such as management classes, a default management class, and copy groups. You can use the COPY MGMTCLASS command to copy a management class and its associated copy groups into the new policy set.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

### Syntax

```
►►—DEFine POLicyset—domainname—setname—[DESCRiption=description]—◄◄
```

### Parameters

#### *domainname*

Specifies the name of the policy domain to which the policy set belongs. This parameter is required.

#### *setname*

Specifies the name you want to assign to the policy set. This parameter is required. The maximum length of this name is 30 characters. You cannot define a policy set named ACTIVE.

#### **DESCRiption**=*description*

Describes the new policy set using a text string. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.

## DEFINE POLICYSET

### Examples

#### Task 1

Define a policy set called VACATION for the EMPLOYEE\_RECORDS policy domain.

**Command:** `define policyset employee_records vacation`

**Output:**

```
ANR1510I Policy set VACATION defined in policy domain
EMPLOYEE_RECORDS.
```

#### Task 2

Define a policy set called SUMMER for the PROG1 policy domain and include a description of Programming Group Policies.

**Command:** `define policyset prog1 summer description="programming group policies"`

**Output:**

```
ANR1510I Policy set SUMMER defined in policy domain PROG1.
```

### Related Commands

Table 49 provides references to related commands.

*Table 49 (Page 1 of 2). Commands Related to DEFINE POLICYSET*

Command	Description	Reference
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
COPY POLICYSET	Creates a copy of an existing policy set.	Page 94
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157

## DEFINE POLICYSET

*Table 49 (Page 2 of 2). Commands Related to DEFINE POLICYSET*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## DEFINE SCHEDULE

---

### DEFINE SCHEDULE (Define a Client or an Administrative Command Schedule)

Use the DEFINE SCHEDULE command to create a new schedule. The DEFINE SCHEDULE command takes two forms, depending on whether the schedule applies to client operations or administrative commands. The syntax and parameters for each form are defined separately.

#### Restrictions:

1. Not all clients can run all scheduled operations, even though ADSM allows you to define the schedule on the server and associate it with the client. For example, a Windows 3.1 client cannot run a schedule when the action is to restore or retrieve files, issue a command, or run an executable script (called a macro by ADSM). An executable script is also known as a command file, a batch file, or a script on different client operating systems. A Macintosh client cannot run a schedule when the action is to restore or retrieve files, or run an executable script.
2. You cannot schedule MACRO or QUERY ACTLOG commands.

---

### Defining Client Schedules

Use the DEFINE SCHEDULE command to create a new schedule for client operations such as backing up or archiving clients in a specified policy domain.

A schedule determines:

- The automatic function to be performed, for example, an incremental backup
- A startup window that indicates a time range during which the operation can begin
- The time period between startup windows

For each schedule, a startup window is specified. The startup window is the time period during which the schedule must be initiated. The schedule will not necessarily complete execution within this window. If the schedule needs to be retried for any reason, the retry attempt must begin before the startup window elapses, or the operation does not restart.

After you define a schedule, use the DEFINE ASSOCIATION command to associate specific clients with the schedule so that they will execute it.

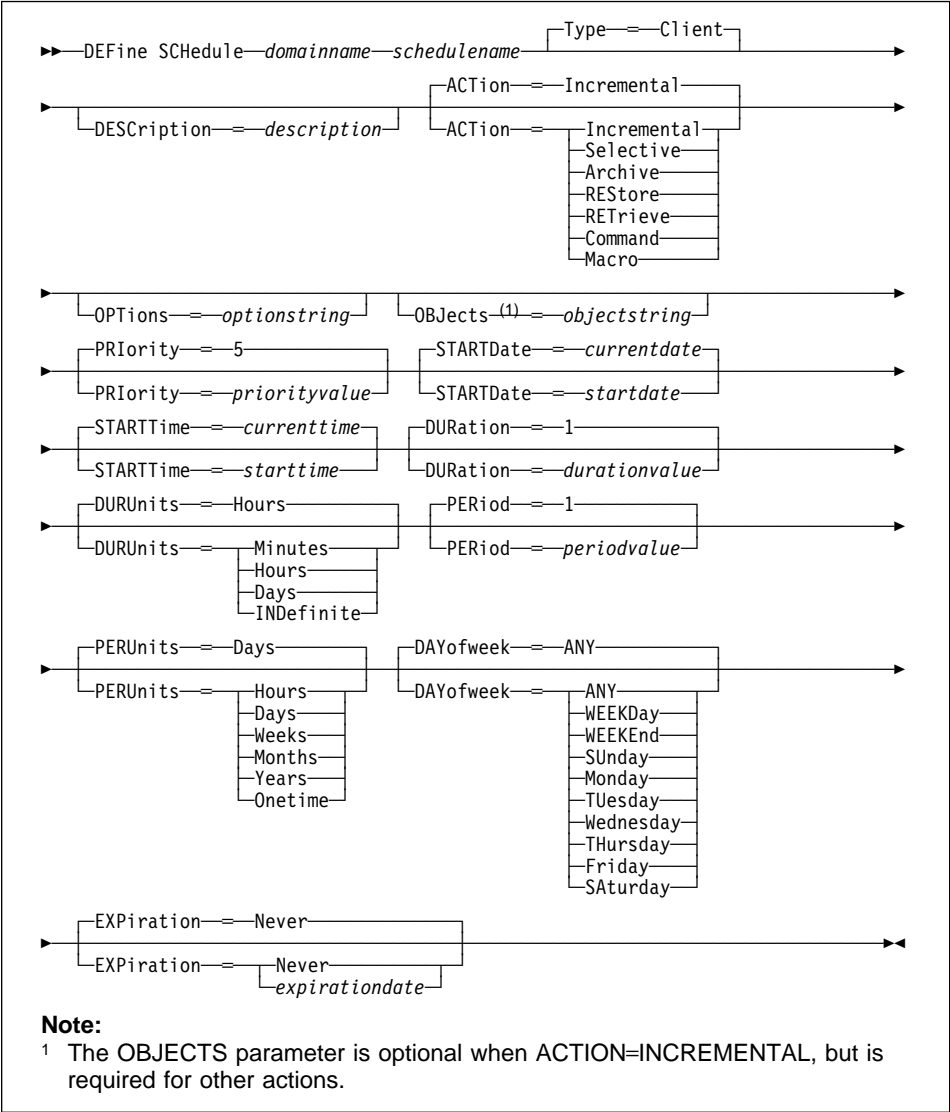
### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the schedule belongs.



# DEFINE SCHEDULE

## Syntax for Defining Client Schedules



## Parameters for Defining Client Schedules

- domainname**  
 Specifies the name of the policy domain to which this schedule belongs. This parameter is required.
- schedulename**  
 Specifies the name of the schedule to be defined. This parameter is required. The maximum length of the name is 30 characters.

## DEFINE SCHEDULE

### **Type=Client**

Specifies that this schedule defines a client schedule. This parameter is optional. The default value is CLIENT. Administrative command schedules are defined by specifying TYPE=ADMINISTRATIVE.

### **DEScRiption=description**

Specifies a description of the schedule. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.

### **ACTioN=actionvalue**

Specifies the action that occurs when this schedule is executed. This parameter is optional. The default value is INCREMENTAL. Possible values are:

#### **Incremental**

Specifies that this schedule initiates the backup of any files that have changed since the last backup and for which the specified time between backups has elapsed. In addition, this schedule initiates the backup of any files for which a backup does not exist. Incremental also backs up any file where all existing backups may have expired.

#### **Selective**

Specifies that this schedule backs up only the files specified with the OBJECTS parameter.

#### **Archive**

Specifies that this schedule archives the files specified with the OBJECTS parameter.

#### **REStore**

Specifies that this schedule restores the files specified with the OBJECTS parameter.

#### **RETRieve**

Specifies that this schedule retrieves the files specified with the OBJECTS parameter.

#### **Command**

Specifies that the action is to be a client operating system command string specified with the OBJECTS parameter.

#### **Macro**

Specifies that a macro whose file name is specified with the OBJECTS parameter is to be executed.

### **OPTioNs=optionstring**

Specifies options that are supplied to the DSMC command at the time the schedule is executed. Only those options that are valid on the DSMC command line may be specified for this parameter. Refer to the appropriate client manual for information on which options are valid from the command line. This parameter is optional.

## DEFINE SCHEDULE

You can specify most options from the client's option file. Each option is preceded by a hyphen and delimited with a space. The option string must be enclosed in quotation marks if it contains any blank characters.

When applicable, these options override the options specified by a client node after it has successfully contacted the server.

Do not include the following options because they have no effect on the client's execution of the scheduled command:

```
MAXCMDRETRIES
QUERYSCHEDPERIOD
RETRYPERIOD
SCHEDLOGNAME
SCHEDMODE
TCPCLIENTADDRESS
TCPCLIENTPORT
```

### **OBjects**=*objectstring*

Specifies the objects for which the specified action is performed. This parameter is required except when ACTION=INCREMENTAL. The allowed objects depend on the action you specify with the ACTION parameter and the operating system used by the clients associated with the schedule.

You can specify more than one object by separating the objects with commas, with no intervening spaces. For example, you can enter a string of file names. If the value for this parameter contains any blank characters, it must be enclosed in quotation marks.

When you specify ACTION=INCREMENTAL and do not specify any objects with this parameter, the DSMC command is invoked without specified objects and attempts to process objects as defined in the client's option file.

### **PRiority**=*priorityvalue*

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer in the range of 1–10, inclusive, with 1 being the highest priority and 10 being the lowest. The default value is 5.

If two or more schedules for a given client have the same window start time, this value determines the order of execution. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

### **STARTDate**=*startdate*

Specifies the date for the beginning of the window in which the schedule is first executed. This parameter is optional. The default value is the current date. This parameter is used in conjunction with the STARTTIME parameter to specify when the initial startup window begins.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the beginning of the startup window in MM/DD/YYYY format (for example, 01/31/1995).

## DEFINE SCHEDULE

### **TODAY**

Specifies that the beginning of the startup window is on the current date.

### **TODAY+numdays or +numdays**

Specifies that the beginning of the startup window is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to specify that the beginning of the initial startup window is 3 days from now, you can specify `STARTDATE=TODAY+3` or `STARTDATE=+3`.

### **STARTTime=starttime**

Specifies the time for the beginning of the window in which the schedule is first executed, in HH:MM format. This parameter is optional. The default value is the current time. This parameter is used in conjunction with the `STARTDATE` parameter to specify when the initial startup window begins.

Possible values are:

#### *HH:MM:SS*

Specifies the time for the beginning of the startup window in HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that the beginning of the startup window is at the current time on the specified start date.

### **NOW+numhours:numminutes or +numhours:numminutes**

Specifies the time for the beginning of the startup window is at the current time plus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the `DEFINE SCHEDULE` command at 5:00 with `STARTTIME=NOW+02:00` or `STARTTIME=+02:00`, the beginning of the initial startup window is at 7:00.

### **NOW-numhours:numminutes or -numhours:numminutes**

Specifies the time for the beginning of the startup window is at the current time minus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the `DEFINE SCHEDULE` command at 5:00 with `STARTTIME=NOW-02:00` or `STARTTIME=-02:00`, the beginning of the initial startup window is at 3:00.

### **DURation=durationvalue**

Specifies the number of units that define the length of the startup window. This parameter is optional. This value must be in the range of 1–999, inclusive. The default value is 1.

This parameter is used in conjunction with the `DURUNITS` parameter to specify the length of the startup window. For example, if you specify `DURATION=20` and `DURUNITS=MINUTES`, the schedule must be started within 20 minutes of the

## DEFINE SCHEDULE

start date and start time. The default value for the length of the startup window is 1 hour. The duration of the window must be shorter than the period between windows.

This value is ignored if you specify `DURUNITS=INDEFINITE`.

### **DURUnits**=*durunitsvalue*

Specifies the time units used to determine the duration of the window in which the schedule can start. This parameter is optional. The default value is `HOURS`.

This parameter is used in conjunction with the `DURATION` parameter to specify how long the startup window remains open to execute the schedule. For example, if `DURATION=20` and `DURUNITS=MINUTES`, the schedule must be started within 20 minutes of the start date and start time. The default value for the length of the startup window is 1 hour. The duration of the window must be shorter than the period between windows. Possible values are:

#### **Minutes**

Specifies that the duration of the window is defined in minutes.

#### **Hours**

Specifies that the duration of the window is defined in hours.

#### **Days**

Specifies that the duration of the window is defined in days.

#### **INDefinite**

Specifies that the startup window has an indefinite duration. The schedule can run any time after the scheduled start time, until the schedule expires.

You cannot specify `DURUNITS=INDEFINITE` unless you also specify `PERUNITS=ONETIME`.

### **PERiod**=*periodvalue*

Specifies the length of time between startup windows for this schedule. This parameter is optional. You can specify an integer in the range of 1–999, inclusive. The default value is 1.

This parameter is used in conjunction with the `PERUNITS` parameter to specify the period between startup windows. For example, if you specify `PERIOD=5` and `PERUNITS=DAYS` (assuming that `DAYOFWEEK=ANY`), the operation is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default value for the length of time between startup windows is 1 day.

This value is ignored if you specify `PERUNITS=ONETIME`.

### **PERUnits**=*perunitsvalue*

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional. The default value is `DAYS`.

This parameter is used in conjunction with the `PERIOD` parameter to specify the period between startup windows. For example, if you specify `PERIOD=5` and `PERUNITS=DAYS` (assuming that `DAYOFWEEK=ANY`), the operation is scheduled every 5 days after the initial start date and start time. The period

## DEFINE SCHEDULE

between startup windows must exceed the duration of each window. The default value for the length of time between startup windows is 1 day. Possible values are:

### **Hours**

Specifies that the time between startup windows for the schedule is in hours.

### **Days**

Specifies that the time between startup windows for this schedule is in days.

### **Weeks**

Specifies that the time between startup windows for the schedule is in weeks.

### **Months**

Specifies that the time between startup windows for the schedule is in months.

### **Years**

Specifies that the time between startup windows for the schedule is in years.

### **Onetime**

Specifies that the schedule executes only once. This value overrides the value specified for the PERIOD parameter.

### **DAYofweek**=*dayofweekvalue*

Specifies the day of the week on which the startup window begins. This parameter is optional. If the beginning of the calculated startup window falls on a day that does not correspond with your value for the DAYOFWEEK parameter, the startup window will be shifted forward in 24-hour increments until the DAYOFWEEK parameter is satisfied.

If you select a value for DAYOFWEEK other than ANY, and depending on the values for PERIOD and PERUNITS, schedules may not be executed when you would expect. Use the QUERY EVENT command to project when schedules will be executed and thereby ensure that the desired result is achieved. The default value is ANY. Possible values are:

### **ANY**

Specifies that the startup window can begin on any day of the week.

### **WEEKDay**

Specifies that the startup window can begin on Monday, Tuesday, Wednesday, Thursday, or Friday.

### **WEEKEnd**

Specifies that the startup window can begin on Saturday or Sunday.

### **SUnday**

Specifies that the startup window begins on Sunday.

## DEFINE SCHEDULE

### **Monday**

Specifies that the startup window begins on Monday.

### **Tuesday**

Specifies that the startup window begins on Tuesday.

### **Wednesday**

Specifies that the startup window begins on Wednesday.

### **Thursday**

Specifies that the startup window begins on Thursday.

### **Friday**

Specifies that the startup window begins on Friday.

### **Saturday**

Specifies that the startup window begins on Saturday.

### **EXPIRATION=expirationdate**

Specifies the date on which this schedule is no longer used, in MM/DD/YYYY format. This parameter is optional. The default value is NEVER. Possible values are:

#### **Never**

Specifies that the schedule never expires.

#### *expirationdate*

Specifies the date on which this schedule is no longer used, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the specified date.

---

## Defining Administrative Command Schedules

Use the DEFINE SCHEDULE command to create a new schedule for the execution of an administrative command.

A schedule determines:

- The administrative command to run
- A startup window that indicates a time range during which the operation can begin
- The time period between startup windows

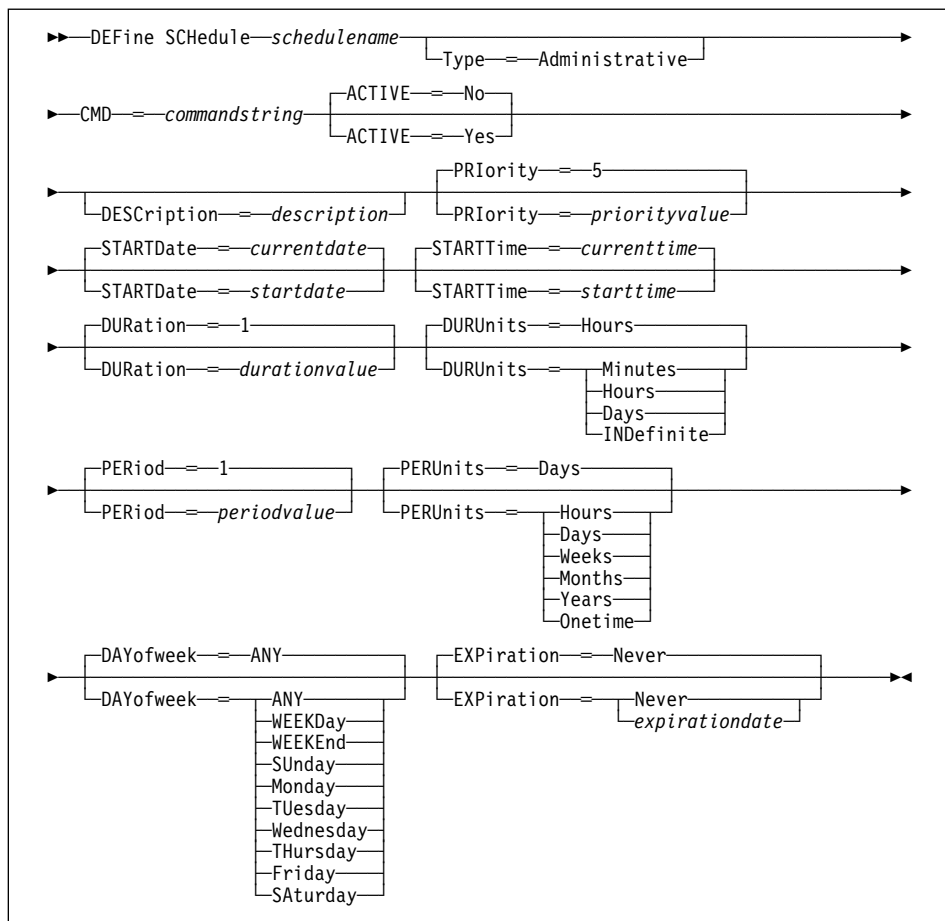
For each schedule, a startup window (or *duration*) is specified. The startup window is the time period during which the schedule must be initiated. The schedule will not necessarily complete execution within this window. If the server is not running when this window starts, but is started before the end of the defined window is reached, the schedule will run when the server is restarted.

## DEFINE SCHEDULE

### Privilege Class

To issue this command, you must have system privilege.

### Syntax for Defining Administrative Command Schedules



### Parameters for Defining Administrative Command Schedules

#### *schedulename*

Specifies the name of the command schedule to be defined. This parameter is required. The maximum length of the name is 30 characters.

#### **Type=Administrative**

Specifies that this schedule defines an administrative command (as opposed to a client command). This parameter is optional. An administrative command is assumed if the CMD parameter is specified.



## DEFINE SCHEDULE

**CMD**=*commandstring*

Specifies the actual administrative command that is to be scheduled. The maximum length of the command is 512 characters. The administrative command must be enclosed in quotation marks if it contains any blank characters. This parameter is required for administrative command schedules.

**ACTIVE**=*activevalue*

Specifies if the defined administrative command is immediately eligible for scheduled execution. This parameter is optional. The default value is NO. The administrative command schedule must be set to the active state with the UPDATE SCHEDULE command to be made eligible for scheduled execution. A defined administrative command schedule will not be executed unless it is set to the active state. Possible values are:

**YES** Specifies that the defined administrative command is eligible for scheduled execution.

**NO** Specifies that the defined administrative command is not eligible for scheduled execution.

**DESCRIPTION**=*description*

Specifies a description of the schedule. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.

**PRIORITY**=*priorityvalue*

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer in the range of 1–10, inclusive, with 1 being the highest priority and 10 being the lowest. The default value is 5.

If two or more administrative schedules have the same window start time, this value determines the order of execution. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

**STARTDATE**=*startdate*

Specifies the date for the beginning of the window in which the schedule is first executed. This parameter is optional. The default value is the current date. This parameter is used in conjunction with the STARTTIME parameter to specify when the initial startup window begins.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the beginning of the startup window in MM/DD/YYYY format (for example, 01/31/1995).

**TODAY**

Specifies that the beginning of the startup window is on the current date.

## DEFINE SCHEDULE

### **TODAY**+*numdays* or +*numdays*

Specifies that the beginning of the startup window is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to specify that the beginning of the initial startup window is 3 days from now, you can specify `STARTDATE=TODAY+3` or `STARTDATE=+3`.

### **STARTTime**=*starttime*

Specifies the time for the beginning of the window in which the schedule is first executed, in HH:MM format. This parameter is optional. The default value is the current time. This parameter is used in conjunction with the `STARTDATE` parameter to specify when the initial startup window begins.

Possible values are:

#### *HH:MM:SS*

Specifies the time for the beginning of the startup window in HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that the beginning of the startup window is at the current time on the specified start date.

### **NOW**+*numhours:numminutes* or +*numhours:numminutes*

Specifies the time for the beginning of the startup window is at the current time plus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the `DEFINE SCHEDULE` command at 5:00 with `STARTTIME=NOW+02:00` or `STARTTIME=+02:00`, the beginning of the initial startup window is at 7:00.

### **NOW**-*numhours:numminutes* or -*numhours:numminutes*

Specifies the time for the beginning of the startup window is at the current time minus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the `DEFINE SCHEDULE` command at 5:00 with `STARTTIME=NOW-02:00` or `STARTTIME=-02:00`, the beginning of the initial startup window is at 3:00.

### **DURation**=*durationvalue*

Specifies the number of units that define the length of the startup window. This parameter is optional. This value must be in the range of 1–999, inclusive. The default value is 1.

This parameter is used in conjunction with the `DURUNITS` parameter to specify the length of the startup window. For example, if you specify `DURATION=20` and `DURUNITS=MINUTES`, the schedule must be started within 20 minutes of the start date and start time. The default value for the length of the startup window is 1 hour. The duration of the window must be shorter than the period between windows.

## DEFINE SCHEDULE

This value is ignored if you specify DURUNITS=INDEFINITE.

### **DURUnits**=*durunitsvalue*

Specifies the time units used to determine the duration of the window in which the schedule can start. This parameter is optional. The default value is HOURS.

This parameter is used in conjunction with the DURATION parameter to specify how long the startup window remains open to execute the schedule. For example, if DURATION=20 and DURUNITS=MINUTES, the schedule must be started within 20 minutes of the start date and start time. The default value for the length of the startup window is 1 hour. The duration of the window must be shorter than the period between windows. Possible values are:

#### **Minutes**

Specifies that the duration of the window is defined in minutes.

#### **Hours**

Specifies that the duration of the window is defined in hours.

#### **Days**

Specifies that the duration of the window is defined in days.

#### **INDefinite**

Specifies that the startup window has an indefinite duration. The schedule can run any time after the scheduled start time, until the schedule expires.

You cannot specify DURUNITS=INDEFINITE unless you also specify PERUNITS=ONETIME.

### **PERiod**=*periodvalue*

Specifies the length of time between startup windows for this schedule. This parameter is optional. You can specify an integer in the range of 1–999, inclusive. The default value is 1.

This parameter is used in conjunction with the PERUNITS parameter to specify the period between startup windows. For example, if you specify PERIOD=5 and PERUNITS=DAYS (assuming that DAYOFWEEK=ANY), the operation is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default value for the length of time between startup windows is 1 day.

This value is ignored if you specify PERUNITS=ONETIME.

### **PERUnits**=*perunitsvalue*

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional. The default value is DAYS.

This parameter is used in conjunction with the PERIOD parameter to specify the period between startup windows. For example, if you specify PERIOD=5 and PERUNITS=DAYS (assuming that DAYOFWEEK=ANY), the operation is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default value for the length of time between startup windows is 1 day. Possible values are:

## DEFINE SCHEDULE

### Hours

Specifies that the time between startup windows for the schedule is in hours.

### Days

Specifies that the time between startup windows for this schedule is in days.

### Weeks

Specifies that the time between startup windows for the schedule is in weeks.

### Months

Specifies that the time between startup windows for the schedule is in months.

### Years

Specifies that the time between startup windows for the schedule is in years.

### Onetime

Specifies that the schedule executes only once. This value overrides the value specified for the PERIOD parameter.

### **DAYofweek**=*dayofweekvalue*

Specifies the day of the week on which the startup window begins. If the beginning of the calculated startup window falls on a day that does not correspond with your value for the DAYOFWEEK parameter, the startup window will be shifted forward in 24-hour increments until the DAYOFWEEK parameter is satisfied.

If you select a value for DAYOFWEEK other than ANY, depending on the values for PERIOD and PERUNITS, schedules may not be executed when you would expect. Use the QUERY EVENT TYPE=ADMINISTRATIVE command to project when schedules will be executed and thereby ensure that the desired result is achieved.

The default value is ANY. Possible values are:

### **ANY**

Specifies that the startup window can begin on any day of the week.

### **WEEKDay**

Specifies that the startup window can begin on Monday, Tuesday, Wednesday, Thursday, or Friday.

### **WEEKEnd**

Specifies that the startup window can begin on Saturday or Sunday.

### **SUnday**

Specifies that the startup window begins on Sunday.

## DEFINE SCHEDULE

### **Monday**

Specifies that the startup window begins on Monday.

### **Tuesday**

Specifies that the startup window begins on Tuesday.

### **Wednesday**

Specifies that the startup window begins on Wednesday.

### **Thursday**

Specifies that the startup window begins on Thursday.

### **Friday**

Specifies that the startup window begins on Friday.

### **Saturday**

Specifies that the startup window begins on Saturday.

### **EXPIRATION=expirationdate**

Specifies the date on which this schedule is no longer used, in MM/DD/YYYY format. This parameter is optional. The default value is NEVER. Possible values are:

### **Never**

Specifies that the schedule never expires.

### *expirationdate*

Specifies the date on which this schedule is no longer used, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the specified date.

## Examples

### **Task 1**

Define a schedule named MONTHLY\_BACKUP that initiates an incremental backup of all associated nodes. Because the specified start date, October 1, 1994 (10/01/1994), is a Thursday and does not match the specified day of the week (Sunday), the initial startup window begins on the first Sunday after October 1, 1994 (10/01/1994). The startup windows for this schedule extend from 01:00 through 03:00. This monthly schedule initiates backup of c: and d: file spaces for all associated nodes.

**Command:** `define schedule standard monthly_backup description="Monthly Backup of c: and d: drives" objects="c:\* d:\*" startdate=10/01/1994 starttime=01:00 duration=2 durunits=hours period=1 perunits=months dayofweek=sunday`

### **Output:**

```
ANR2500I Schedule MONTHLY_BACKUP defined in policy domain
STANDARD.
```

## DEFINE SCHEDULE

### Task 2

Define a schedule named WEEKLY\_BACKUP that initiates an incremental backup of all associated nodes. The initial startup window for this schedule extends from 23:00 on Saturday, July 4, 1994 (07/04/1994), to 03:00 on Sunday, July 5, 1994 (07/05/1994). Subsequent windows begin at 23:00, every Saturday. No messages are returned to the client node when this schedule is executed.

**Command:** `define schedule employee_records weekly_backup  
startdate=07/04/1994 starttime=23:00 duration=4 durunits=hours  
perunits=weeks dayofweek=saturday options=-quiet`

#### Output:

```
ANR2500I Schedule WEEKLY_BACKUP defined in policy domain  
EMPLOYEE_RECORDS.
```

### Task 3

Define a schedule named BACKUP\_ARCHIVEPOOL that backs up the primary storage pool ARCHIVEPOOL to the copy storage pool RECOVERYPOOL. The backup runs at 8 p.m. every two days.

**Command:** `define schedule backup_archivepool type=administrative  
cmd="backup stgpool archivepool recoverypool" active=yes  
starttime=20:00 period=2`

#### Output:

```
ANR2577I Schedule BACKUP_ARCHIVEPOOL defined.
```

## Related Commands

Table 50 provides references to related commands.

Command	Description	Reference
COPY SCHEDULE	Creates a copy of an existing schedule.	Page 96
DEFINE ASSOCIATION	Associates one or more clients with a schedule.	Page 100
DELETE SCHEDULE	Deletes a schedule from the database.	Page 229
QUERY EVENT	Displays information about scheduled and completed events for selected clients.	Page 362

## DEFINE SCHEDULE

*Table 50 (Page 2 of 2). Commands Related to DEFINE SCHEDULE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY SCHEDULE	Displays information about one or more schedules.	Page 422
SET MAXCMDRETRIES	Specifies the maximum number of retries after a failed attempt to execute a scheduled command.	Page 517
SET RETRYPERIOD	Specifies the time between retry attempts by the client scheduler.	Page 530
UPDATE SCHEDULE	Changes the attributes of a schedule.	Page 590

## DEFINE STGPOOL

---

### DEFINE STGPOOL (Define a Storage Pool)

Use the DEFINE STGPOOL command to name a set of volumes as a primary storage pool or a copy storage pool. The volumes in a pool belong to the same device class. A primary storage pool is used as the destination for backup files, archive files, or files migrated from client nodes. A copy storage pool is used to store backup copies of files that are in primary storage pools.

During installation, ADSM provides three predefined random access storage pools:

<b>BACKUPPOOL</b>	A storage destination for workstation files that are backed up to the server
<b>ARCHIVEPOOL</b>	A storage destination for workstation files that are archived to the server
<b>SPACEMGPOOL</b>	A storage destination for files migrated from client nodes

Use this command to increase ADSM data storage capacity by defining additional storage pools. You can define storage pools as necessary to manage your storage.

You can restrict administrator authority, by storage pool, to control administrator access to your data storage.

If you want to migrate files from one primary storage pool to another, you must specify the NEXTSTGPOOL parameter. This parameter links primary storage pools together into a hierarchy for data storage.

To determine the name of the device class (and the corresponding device type) to which you want the storage pool assigned, use the QUERY DEVCLASS command.

After you define a storage pool, you can use the DEFINE VOLUME command to assign volumes to the storage pool.

The DEFINE STGPOOL command takes three forms, depending on whether you are defining a primary storage pool assigned to random access devices, a primary storage pool assigned to sequential access devices, or a copy storage pool. The syntax and parameters for each form are defined separately.

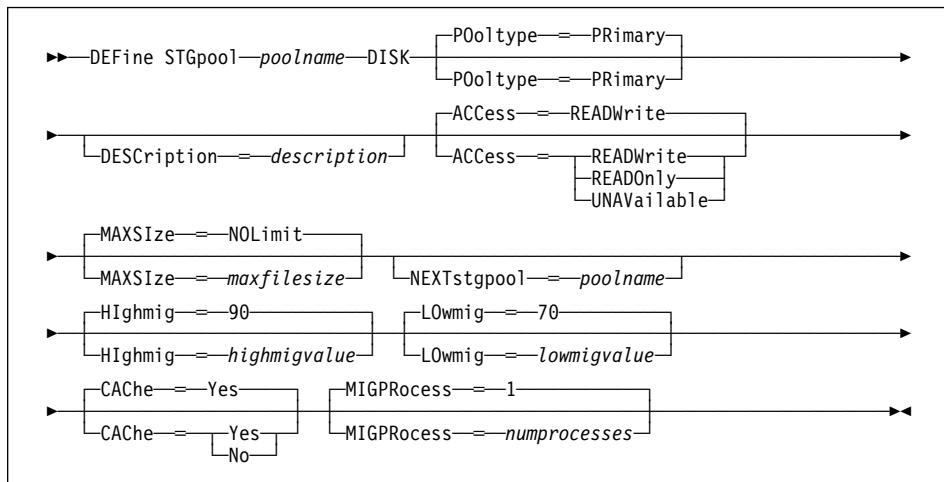
### Privilege Class

To issue this command, you must have system privilege.



## DEFINE STGPOOL

### Syntax for Defining Random Access Storage Pools



### Parameters for Defining a Random Access Storage Pool

#### *poolname*

Specifies the name of the storage pool to be defined. This parameter is required. The maximum length of the name is 30 characters.

#### **DISK**

Specifies that you want to define a storage pool to the DISK device class (the DISK device class is predefined during installation). This parameter is required.

#### **POOLtype=PRIMARY**

Specifies that you want to define a primary storage pool. This parameter is optional. The default value is PRIMARY.

#### **DESCRIPTION=description**

Specifies a description of the storage pool to be defined. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.

#### **ACCESS=accessmode**

Specifies how users and system processes (like migration and reclamation) can access files in the storage pool. This parameter is optional. The default value is READWRITE. Possible values are:

#### **READWrite**

Specifies that users and system processes (like migration) can read and write to files stored on volumes in the storage pool.

#### **READOnly**

Specifies that users can only read files from the volumes in the storage pool.

## DEFINE STGPOOL

No new writes are permitted to the volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *readonly*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool.

### UNAVailable

Specifies that users cannot access files stored on volumes in the storage pool.

No new writes are permitted to volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool. Files can also be moved or copied from this storage pool to another storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *unavailable*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool.

### MAXSize=*maxfilesize*

Specifies the maximum file size for the storage pool. This parameter is optional. The value of this parameter can be either NOLIMIT, or an integer followed by a scale factor. The default value is NOLIMIT, which specifies that there is no maximum size limit for files stored in the storage pool.

To limit the maximum file size, specify an integer (1–999999) followed by a scale factor. Scale factors and their meanings are:

Scale Factor	Meaning
K	Kilobyte
M	Megabyte
G	Gigabyte
T	Terabyte

For example, MAXSIZE=5G specifies that the maximum file size for this storage pool is 5 gigabytes.

If a file exceeds the maximum size and no pool is specified as the NEXTSTGPOOL, the file is not stored. If a file exceeds the maximum size and a pool is specified as the NEXTSTGPOOL, the file is stored in the next storage pool that can accept the file size. If you specify the NEXTSTGPOOL parameter, at least one storage pool in your hierarchy should have a MAXSIZE value of NOLIMIT. By setting MAXSIZE to NOLIMIT for at least one pool, you ensure that no matter what its size, a file can be stored somewhere.

## DEFINE STGPOOL

### **NEXTstgpool**=*poolname*

Specifies the name of the storage pool to which files can be migrated. This parameter is optional.

A pool specified with the NEXTSTGPOOL parameter must be a primary storage pool. A copy storage pool cannot be specified.

If you do not specify a value for this parameter, migration cannot occur and files that exceed the value specified for the MAXSIZE parameter cannot be placed in another storage pool.

You cannot create a chain of storage pools that leads to an endless loop through the NEXTSTGPOOL parameter. At least one storage pool in the hierarchy must have no value specified for NEXTSTGPOOL.

### **Highmig**=*highmigvalue*

Specifies when ADSM can start migration for this storage pool, as a percentage of the storage pool capacity. This parameter is optional. You can specify an integer from 0–100, inclusive. The default value is 90 percent.

When the storage pool occupancy exceeds the value for this parameter, ADSM can initiate migration, by node, to the next storage pool, as defined with the NEXTSTGPOOL parameter. You can specify HIGHMIG=100 to prevent migration for this storage pool (assuming you have defined the NEXTSTGPOOL parameter).

### **LOWmig**=*lowmigvalue*

Specifies when ADSM can stop migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 0–99. The default value is 70.

When the storage pool occupancy reaches the value for this parameter, ADSM will not initiate migration of another node's files. Because all file spaces that belong to a node are migrated together, the occupancy of the storage pool can fall below the value you specified for this parameter. You can set LOWMIG=0 to permit migration to empty the storage pool.

### **CAChe**=*cachingvalue*

For random access storage pools only, specifies whether a cached copy of a file that has been migrated remains in the random access storage pool. This parameter is optional. The primary copy of the file is migrated to the next storage pool, as defined with the NEXTSTGPOOL parameter.

Caching improves the retrievability of files that have been migrated to sequential access storage pools by permitting cached copies of files to remain on the storage pool. The cached copies are available, if needed, or can be overwritten when necessary. The default value is YES. Possible values are:

#### **Yes**

Specifies that caching is enabled.

#### **No**

Specifies that caching is disabled.

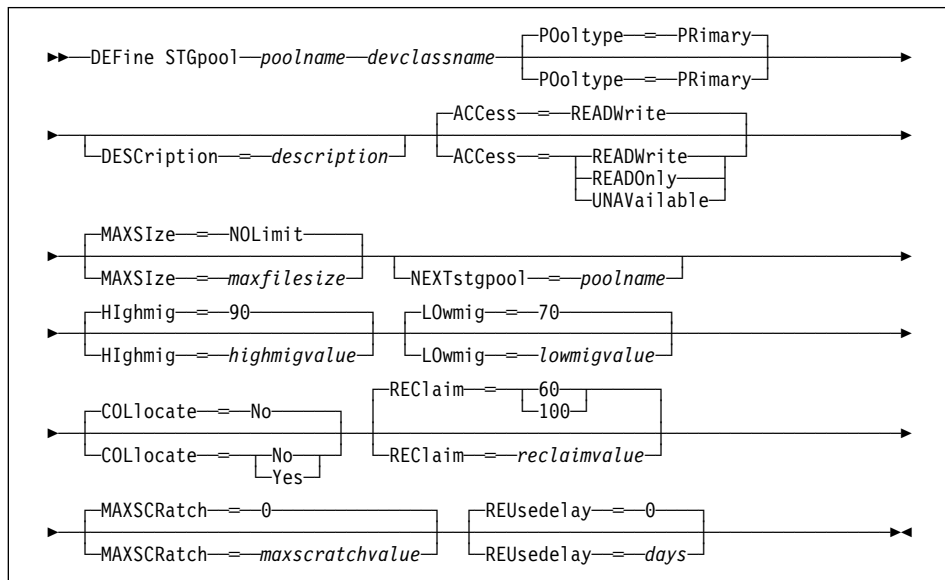
## DEFINE STGPOOL

### **MIGProcesses=numprocesses**

Specifies the number of processes that are used for migrating files from this storage pool. This parameter is optional. You can specify an integer from 1–999, inclusive. The default value is 1.

During migration, these processes are performed in parallel to provide the potential for improved migration rates.

## Syntax for Defining Primary Sequential Access Storage Pools



## Parameters for Defining a Primary Sequential Access Storage Pool

### *poolname*

Specifies the name of the storage pool to be defined. This parameter is required. The maximum length of the name is 30 characters.

### *devclassname*

Specifies the name of the device class to which this storage pool is assigned. This parameter is required. You can specify any device class defined with the DEFINE DEVCLASS command.

### **POoltype=PRimary**

Specifies that you want to define a primary storage pool. This parameter is optional. The default value is PRIMARY.

### **DESCRiption=description**

Specifies a description of the storage pool to be defined. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.

## DEFINE STGPOOL

### **ACCess**=*accessmode*

Specifies how users and system processes (like migration and reclamation) can access files in the storage pool. This parameter is optional. The default value is READWRITE. Possible values are:

#### **READWrite**

Specifies that users and system processes (like migration) can read and write to files stored on volumes in the storage pool.

#### **READOnly**

Specifies that users can only read files stored on the volumes in the storage pool.

No new writes are permitted to the volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *readonly*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool.

#### **UNAVailable**

Specifies that users cannot access files stored on volumes in the storage pool.

No new writes are permitted to volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool. Files can also be moved or copied from this storage pool to another storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *unavailable*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool.

### **MAXSize**=*maxfilesize*

Specifies the maximum file size for the storage pool. This parameter is optional. The value of this parameter can be either NOLIMIT, or an integer followed by a scale factor. The default value is NOLIMIT, which specifies that there is no maximum size limit for files stored in the storage pool.

To limit the maximum file size, specify an integer (1–999999) followed by a scale factor. Scale factors and their meanings are:

<b>Scale Factor</b>	<b>Meaning</b>
K	Kilobyte
M	Megabyte
G	Gigabyte
T	Terabyte

## DEFINE STGPOOL

For example, `MAXSIZE=5G` specifies that the maximum file size for this storage pool is 5 gigabytes.

If a file exceeds the maximum size and no pool is specified as the `NEXTSTGPOOL`, the file is not stored. If a file exceeds the maximum size and a pool is specified as the `NEXTSTGPOOL`, the file is stored in the next storage pool that can accept the file size. If you specify the `NEXTSTGPOOL` parameter, at least one storage pool in your hierarchy should have a `MAXSIZE` value of `NOLIMIT`. By setting `MAXSIZE` to `NOLIMIT` for at least one pool, you ensure that no matter what its size, a file can be stored somewhere.

### **NEXTstgpool**=*poolname*

Specifies the name of the storage pool to which files can be migrated. This parameter is optional.

A pool specified with the `NEXTSTGPOOL` parameter must be a primary storage pool. A copy storage pool cannot be specified.

If you do not specify a value for this parameter, migration cannot occur from this storage pool and files that exceed the value specified for the `MAXSIZE` parameter cannot be placed in another storage pool.

You cannot create a chain of storage pools that leads to an endless loop through the `NEXTSTGPOOL` parameter. At least one storage pool in the hierarchy must have no value specified for `NEXTSTGPOOL`.

### **Hghmig**=*highmigvalue*

Specifies when ADSM can start migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 1–100, inclusive. The default value is 90 percent.

When the storage pool occupancy exceeds the value for this parameter, ADSM can initiate migration, by volume, to the next storage pool, as defined with the `NEXTSTGPOOL` parameter. You can specify `HIGHMIG=100` to prevent migration for this storage pool (assuming you have defined the `NEXTSTGPOOL` parameter).

### **Lowmig**=*lowmigvalue*

Specifies when ADSM can stop migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 0–99. The default value is 70.

When the storage pool occupancy reaches the value for this parameter, ADSM will not initiate migration of files from another volume. Because all files that are on a single volume are migrated together, the occupancy of the storage pool can fall below the value you specified for this parameter. You can set `LOWMIG=0` to permit migration to empty the storage pool.

### **COLlocate**=*collocation*

Specifies whether ADSM attempts to keep file spaces assigned to as few volumes as possible. This parameter is optional. The default value is `NO`.

Collocation reduces the number of sequential access media mounts for restore, retrieve, and recall operations. However, collocation increases both the amount

## DEFINE STGPOOL

of server time needed to collocate files for storing, and the number of volumes required. Collocation may not be useful on systems with only one or two sequential devices. For additional information, see *ADSM Administrator's Guide*.

Possible values are:

**Yes**

Specifies that collocation is enabled.

**No**

Specifies that collocation is disabled.

**REClaim**=*reclaimvalue*

Specifies when to initiate reclamation of fragmented space on volumes in the storage pool, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space usable again by moving any remaining active files from one volume to another volume, thus making the original volume available for reuse. This parameter is optional. You can specify an integer from 1–100, inclusive. The default value is 60.

Space becomes fragmented as files on a volume are deleted because of processes such as expiration or file space deletion. The amount of space that can be reclaimed on a volume increases as files on the volume are deleted. When the percentage of space that can be reclaimed on a volume rises above the value for the RECLAIM parameter, the volume is reclaimed. Active files on the volume are rewritten to other volumes in the storage pool, making the original volume available for new files.

It is recommended that you specify a value of 50 percent or greater for this parameter so that files stored on two volumes can be combined onto a single output volume.

**MAXSCRatch**=*maxscratchvalue*

Specifies the maximum number of scratch volumes that can be requested by ADSM. This parameter is optional. You can specify an integer from 0–100000000, inclusive. The default value is 0.

The value specified for this parameter is used to estimate the total number of volumes available in the storage pool, and the corresponding estimated capacity for the storage pool.

If you do not specify a value for this parameter, no scratch media are requested by ADSM.

Scratch volumes are automatically deleted from the storage pool when they become empty. When scratch volumes with the device type of FILE are deleted, the space that the volumes occupied is freed by ADSM and returned to the file system.

**REUsedelay**=*days*

Specifies the number of days that must elapse after all files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. This parameter is optional. You can specify an integer from 0–9999, inclusive. The default value is 0, which means that a volume can be rewritten or

## DEFINE STGPOOL

returned to the scratch pool as soon as all files have been deleted from the volume.

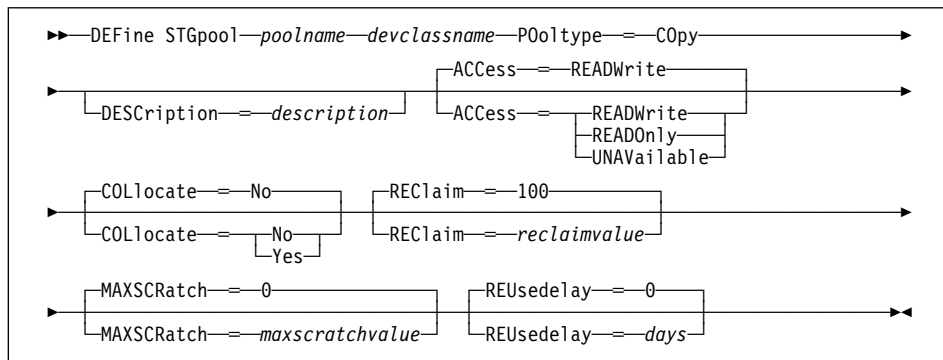
By specifying this parameter, you can ensure that the database could be restored to an earlier level and database references to files in the storage pool would still be valid.

### Usage Notes

The migration occupancy represented by the HIGHMIG and LOWMIG parameter is computed differently for random and sequential access storage pools:

- For random access storage pools, the occupancy is computed as a percentage of the total byte capacity of the storage pool.
- For sequential access storage pools, the occupancy is based on the number of volumes in the storage pool that contain nondeleted files.

### Syntax for Defining Copy Storage Pools



### Parameters for Defining a Copy Storage Pool

#### *poolname*

Specifies the name of the copy storage pool to be defined. This parameter is required. The maximum length of the name is 30 characters.

#### *devclassname*

Specifies the name of the sequential access device class to which this copy storage pool is assigned. This parameter is required. You can specify any device class except DISK.

#### **POOLTYPE=COPY**

Specifies that you want to define a copy storage pool. This parameter is required.

#### **DESCRIPTION=description**

Specifies a description of the copy storage pool to be defined. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. The default is no description.



## DEFINE STGPOOL

### **ACCess**=*accessmode*

Specifies how users and system processes (like reclamation) can access files in the copy storage pool. This parameter is optional. The default value is READWRITE. Possible values are:

#### **READWrite**

Specifies that files can be read from and written to the volumes in the copy storage pool.

#### **READOnly**

Specifies that users can only read files stored on the volumes in the copy storage pool.

No new writes are permitted to the volumes in the copy storage pool from other volumes outside the storage pool. Files cannot be backed up to the copy storage pool, but existing backup copies can be restored. System processes (like reclamation) are permitted to move files within the volumes in the copy storage pool.

#### **UNAVailable**

Specifies that users cannot access files stored on volumes in the copy storage pool.

No new writes are permitted to volumes in the copy storage pool from other volumes outside the storage pool. Files cannot be backed up to the copy storage pool, but existing backup copies can be restored. System processes (like reclamation) are permitted to move files within the volumes in the copy storage pool.

### **COLlocate**=*collocation*

Specifies whether ADSM attempts to keep file spaces assigned to as few volumes as possible. This parameter is optional. The default value is NO.

Collocation reduces the number of sequential access media mounts for restore, retrieve, and recall operations. However, collocation increases both the amount of server time needed to collocate files for storing, and the number of volumes required. Collocation may not be useful on systems with only one or two sequential devices. For additional information, see *ADSM Administrator's Guide*.

Possible values are:

#### **Yes**

Specifies that collocation is enabled.

#### **No**

Specifies that collocation is disabled.

### **REClaim**=*reclaimvalue*

Specifies when to initiate reclamation of fragmented space on volumes in the copy storage pool, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space usable again by moving any remaining active files from one volume to another volume, thus making the original volume available for reuse. This parameter is optional. You can specify an integer from

## DEFINE STGPOOL

1–100, inclusive. The default value is 100, which means that reclamation is not performed.

The amount of space that can be reclaimed on a volume increases as files on the volume are deleted because of processes such as expiration or file space deletion. It is recommended that you specify a value of 50 percent or greater for this parameter so that files stored on two volumes can be combined onto a single output volume.

For offsite storage pool volumes, when a copy pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to retrieve the active files on the reclaimable volume from a primary or copy storage pool that is onsite, and then write these files to an available volume in the original copy storage pool. Effectively, these files are moved back to the onsite location, although they could be obtained from the offsite volume after a disaster if a database backup is used which references the files on the offsite volume. Because of the way reclamation works with offsite volumes, use it carefully with copy storage pools.

### **MAXSCRatch**=*maxscratchvalue*

Specifies the maximum number of scratch volumes that can be requested by ADSM. This parameter is optional. You can specify an integer from 0–100000000, inclusive. The default value is 0.

The value specified for this parameter is used to estimate the total number of volumes available in the copy storage pool, and the corresponding estimated capacity for the copy storage pool.

If you do not specify a value for this parameter, no scratch media are requested by ADSM.

Scratch volumes are automatically deleted from the storage pool when they become empty. However, if the access mode for a scratch volume is OFFSITE, the volume will not be deleted from the copy storage pool until the access mode is changed; this allows an administrator to query ADSM for empty, offsite scratch volumes and return these to the onsite location.

When scratch volumes with the device type of FILE become empty and are deleted, the space that the volumes occupied is freed by ADSM and returned to the file system.

### **REUsedelay**=*days*

Specifies the number of days that must elapse after all files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. This parameter is optional. You can specify an integer from 0–9999, inclusive. The default value is 0, which means that a volume can be rewritten or returned to the scratch pool as soon as all files have been deleted from the volume.

**Important:** Specifying this parameter, you can ensure that the database can be restored to an earlier level and database references to files in the copy storage pool are still valid. For this to happen, you must set this parameter to a value

## DEFINE STGPOOL

greater than the number of days you plan on retaining the oldest database backup. For additional information, refer to *ADSM Administrator's Guide*.

### Examples

#### Task 1

Define a primary storage pool, POOL1, to the DISK device class, with caching enabled. The maximum file size is limited to 5MB, with a high migration threshold of 70 percent, and a low migration threshold of 30 percent. Any files larger than 5MB are stored in subordinate storage pools beginning with the BACKUPPOOL storage pool.

**Command:** `define stgpool pool1 disk description="disk stgpool" maxsize=5m highmig=70 lowmig=30 cache=Yes nextstgpool=backuppool`

#### Output:

```
ANR2200I Storage pool POOL1 defined (device class DISK).
```

#### Task 2

Define a primary storage pool named 8MMPPOOL to the 8MMTAPE device class (with a device type of 8MM) with a maximum file size of 5MB. Any files larger than 5MB are stored in subordinate pools, beginning with POOL1. Collocation is enabled for the storage pool. As many as 5 scratch volumes can be requested for this storage pool.

**Command:** `define stgpool 8mmpool 8mmtape maxsize=5m nextstgpool=pool1 collocate=yes maxscratch=5`

#### Output:

```
ANR2200I Storage pool 8MMPPOOL defined (device class 8MMTAPE).
```

#### Task 3

Define a copy storage pool, COPYSTG, to the PLAINFILES device class. Include "Copy Storage Pool" as part of the description for this pool.

**Command:** `define stgpool copystg plainfiles pooltype=copy description="copy storage pool"`

## DEFINE STGPOOL

### Output:

```
ANR2200I Storage pool COPYSTG defined (device class  
PLAINFILES).
```

### Related Commands

Table 51 provides references to related commands.

Table 51. Commands Related to DEFINE STGPOOL

Command	Description	Reference
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE STGPOOL	Deletes a storage pool from data storage.	Page 232
QUERY DEVCLASS	Displays information about one or more device classes.	Page 350
QUERY STGPOOL	Displays information about one or more storage pools.	Page 436
RESTORE STGPOOL	Restores files to a primary storage pool from copy storage pools.	Page 488
RESTORE VOLUME	Restores files stored on specified volumes in a primary storage pool from copy storage pools.	Page 493
UPDATE STGPOOL	Changes the attributes of a storage pool.	Page 605

## DEFINE VOLUME

### DEFINE VOLUME (Define a Volume in a Storage Pool)

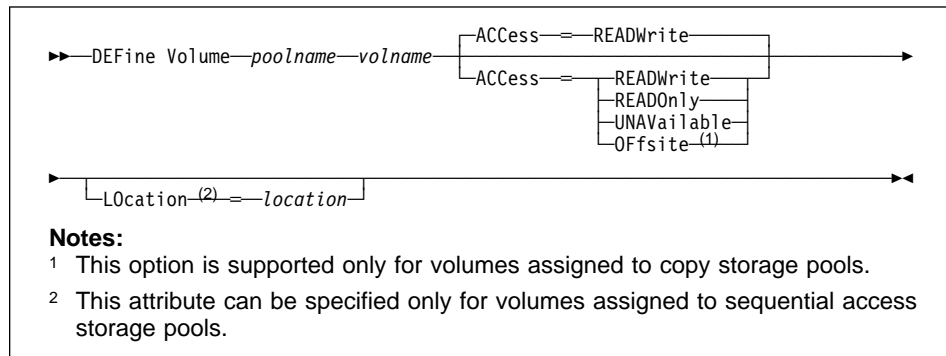
Use the DEFINE VOLUME command to assign a random or sequential access volume to be used for storage within an existing storage pool. You can define a volume to either a primary storage pool or a copy storage pool. You must define each volume to be used in a storage pool, unless you allow scratch volumes for the storage pool.

**Attention:** The size of an allocated database, recovery log, or storage pool volume cannot be changed once it has been defined to the ADSM server. ADSM uses the initial size allocation of the volume at the time it is defined to the server to calculate data placement for later retrieval.

### Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool to which the volume is assigned.

### Syntax



### Parameters

#### *poolname*

Specifies the name of the storage pool to which the volume is assigned. This parameter is required.

#### *volname*

Specifies the name of the storage pool volume to be defined. This parameter is required. The volume name that you can specify depends on the device class device type of the associated storage pool:

- For device classes with a FILE or DISK device type — Specify the file name of the file containing the volume data. This file name can be entered with either the fully qualified path name or the path name relative to the directory identified in the DIRECTORY parameter of the associated device class. For example, if the file name is mainvol, the path name to be entered is:

```
c:\admserv\mainvol
```

## DEFINE VOLUME

- For device classes with an 4MM, 8MM, DLT, or QIC device types—Specify a 1 to 6 alphanumeric volume name. An example is DSMT01 (ADSM always converts volume names to upper-case).
- For device classes with a LAN3995 device type —Specify the name of one or both side of an optical cartridge. For example, any of the following is acceptable:
  - side1
  - side2
  - side1/side2
  - side2/side1
- For device classes with an OPTICAL or MO35 device type—Specify a 1 to 32 alphanumeric volume name.
- For device classes with a REMOVABLEFILE device type—Specify a 1 to 8 alphanumeric volume name, with no embedded blanks or periods.

No volume name used by ADSM can contain any embedded blanks or equal signs.

### **ACCess**=*accessmode*

Specifies how users and system processes (like migration) can access files in the storage pool volume. This parameter is optional. The default value is READWRITE. Possible values are:

#### **READWrite**

Specifies that users and system processes (like migration) can read from and write to files stored on the volume.

#### **READOnly**

Specifies that users and system processes (like migration) can only read files stored on the volume.

#### **UNAVailable**

Specifies that neither users nor system processes (like migration) can access files stored on the volume.

If you define a random access volume as *unavailable*, you cannot vary the volume online. If you define a sequential access volume as *unavailable*, ADSM does not attempt to access the volume.

#### **OFFsite**

Specifies that the volume is at an offsite location from which it cannot be mounted. This option can only be specified for volumes in copy storage pools.

Use this parameter to help you track volumes at offsite locations. ADSM treats volumes designated as offsite differently:

- Mount requests are not generated for volumes assigned to offsite locations

## DEFINE VOLUME

- Data can be reclaimed or moved from offsite volumes by retrieving files from other storage pools
- Empty, offsite scratch volumes are not deleted from the copy storage pool

### **Location**=*location*

Specifies the location of the volume. This parameter is optional. It can be specified only for volumes in sequential access storage pools. The location information can be a maximum length of 255 characters. The location must be enclosed in quotation marks if it contains any blank characters. The default is no location.

## Usage Notes

For sequential access storage pools, it is not necessary to manually define any volumes. You can use the MAXSCRATCH parameter to permit ADSM to mount its own scratch volumes.

## Examples

### Task 1

A storage pool named POOL1 is assigned to a device class with a device type of 8MM. Define a volume named mainvol to this storage pool, with READWRITE access.

**Command:** `define volume pool1 mainvol access=readwrite`

### Output:

```
ANR2206I Volume MAINVOL defined in storage pool POOL1
(device class 8MM).
```

## DEFINE VOLUME

### Task 2

A storage pool named 8MMPPOOL is assigned to a device class with a device type of 8MM. Define a volume named TAPE01 to this storage pool.

**Command:** `define volume 8mmpool tape01`

**Output:**

```
ANR2206I Volume TAPE01 defined in storage pool 8MMPPOOL
(device class 8MM).
```

### Task 3

A storage pool named OPTPOOL is assigned to a device class with a device type of LAN3995. Define a volume named glm3a to this storage pool.

**Command:** `define volume optpool glm3a`

**Output:**

```
ANR2206I Volume GLM3A/GLM3B defined in storage pool OPTPOOL
(device class LAN3995).
```

## Related Commands

Table 52 provides references to related commands.

*Table 52. Commands Related to DEFINE VOLUME*

Command	Description	Reference
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
QUERY VOLUME	Displays information about one or more storage pool volumes.	Page 450
UPDATE VOLUME	Updates the attributes of one or more storage pool volumes.	Page 619



## DELETE ASSOCIATION

---

### DELETE ASSOCIATION (Delete Node Associations with a Schedule)

Use the DELETE ASSOCIATION command to remove the association between one or more clients and a schedule.

This command does not delete a client from ADSM. It only removes the association of the client from the specified schedule. When you delete the association of a client to a schedule, the client's data is no longer managed by the specified schedule.

If you try to disassociate a client from a schedule to which it is not associated, this command will have no effect for that client.

#### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the domain to which the schedule belongs.

#### Syntax

```
►—DELEte ASSOCiation—domainname—schedulename—nodename—◄
```

#### Parameters

*domainname*

Specifies the name of the policy domain to which the schedule belongs. This parameter is required.

*schedulename*

Specifies the name of the schedule from which clients are to be disassociated. This parameter is required.

*nodename*

Specifies the name of the node that is no longer associated with the schedule. This parameter is required. You can specify a list of clients which are to be no longer associated with the specified schedule. The items in the list are separated by commas, with no intervening spaces. You can also use a pattern-matching expression to specify a name. All matching clients are disassociated from the specified schedule.

#### Examples

##### Task 1

Remove the association between the node named JEFF, assigned to the DOMAIN1 policy domain, and the WEEKLY\_BACKUP schedule.

## DELETE ASSOCIATION

**Command:** `delete association domain1 weekly_backup jeff`

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR2511I Node JEFF disassociated from schedule WEEKLY_BACKUP  
in policy domain DOMAIN1.
```

### Task 2

Remove the association between selected clients (assigned to the DOMAIN1 policy domain) and the WEEKLY\_BACKUP schedule so that this schedule is no longer executed by these clients. The nodes that are disassociated from the schedule contain ABC or XYZ in the node name.

**Command:** `delete association domain1 weekly_backup *abc*,*xyz*`

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR2511I Node XABCEF disassociated from schedule  
WEEKLY_BACKUP in policy domain DOMAIN1.  
ANR2511I Node ABCONTROL disassociated from schedule  
WEEKLY_BACKUP in policy domain DOMAIN1.  
ANR2511I Node MARTXYZ disassociated from schedule  
WEEKLY_BACKUP in policy domain DOMAIN1.
```

## Related Commands

Table 53 provides references to related commands.

*Table 53. Commands Related to DELETE ASSOCIATION*

Command	Description	Reference
DEFINE ASSOCIATION	Associates one or more clients with a schedule.	Page 100
QUERY ASSOCIATION	Displays the clients associated with one or more schedules.	Page 322

## DELETE COPYGROUP

---

### DELETE COPYGROUP (Delete a Copy Group)

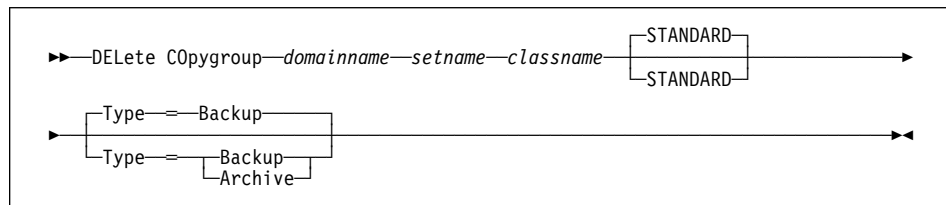
Use the DELETE COPYGROUP command to delete an existing backup or archive copy group from a specified management class. A copy group in the ACTIVE policy set cannot be deleted.

Any files that are bound to a deleted copy group are managed by the default management class, when the policy set is activated.

#### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the copy group belongs.

#### Syntax



#### Parameters

##### *domainname*

Specifies the name of the policy domain to which the copy group to be deleted belongs. This parameter is required.

##### *setname*

Specifies the name of policy set to which the copy group to be deleted belongs. This parameter is required.

##### *classname*

Specifies the name of the management class to which the copy group to be deleted belongs. This parameter is required.

##### **STANDARD**

Specifies the name of the copy group, STANDARD. This parameter is optional. If you do not specify this value, the command defaults to STANDARD.

##### **Type=copygroup type**

Specifies whether a backup or archive copy group is to be deleted. This parameter is optional. The default value is BACKUP. Possible values are:

##### **Backup**

Specifies that a backup copy group is to be deleted.

##### **Archive**

Specifies that an archive copy group is to be deleted.

## DELETE COPYGROUP

### Usage Notes

You can delete the predefined STANDARD copy group located in the STANDARD policy domain. If, however, at a later time you reinstall the ADSM server, all STANDARD policy objects are restored. Also, if after the server reinstall you are reestablishing your server database (via the database salvage commands), during DSMSEV AUDITDB command processing messages are issued indicating that an instance count does not agree with actual data. DSMSEV AUDITDB does not delete the STANDARD policy objects; however, it does synchronize the instance count to include the restored STANDARD policy objects.

### Examples

#### Task 1

Delete the backup copy group, STANDARD, from the ACTIVEFILES management class that is assigned to the VACATION policy set of the EMPLOYEE\_RECORDS policy domain.

**Command:** `delete copygroup employee_records vacation activefiles`

#### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR1531I Backup copy group STANDARD deleted from  
policy domain EMPLOYEE_RECORDS, set VACATION,  
management class ACTIVEFILES.
```

#### Task 2

Delete the archive copy group, STANDARD, from the MCLASS1 management class that is assigned to the SUMMER policy set of the PROG1 policy domain.

**Command:** `delete copygroup prog1 summer mclass1 Type=Archive`

#### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR1536I Archive copy group STANDARD deleted from policy  
domain PROG1, set SUMMER, management class MCLASS1.
```

## DELETE COPYGROUP

### Related Commands

Table 54 provides references to related commands.

*Table 54. Commands Related to DELETE COPYGROUP*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539

## DELETE DBBACKUPTRIGGER

---

### DELETE DBBACKUPTRIGGER (Delete the Database Backup Trigger)

Use the DELETE DBBACKUPTRIGGER command to delete the definition for the database backup trigger.

When you delete the definition for the database backup trigger, ADSM does not run backups of the ADSM database automatically and does not remove unnecessary log records from the recovery log. After you delete the database backup trigger, be sure to use the SET LOGMODE command to change the log mode to NORMAL.

#### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

#### Syntax

```
▶▶—DELeTe DBBackuptrigger—▶▶
```

#### Parameters

None.

#### Examples

##### Task

Delete the definition for the database backup trigger.

**Command:** delete dbbackuptrigger

##### Output:

```
ANR2286I Database backup trigger deleted.
```

## DELETE DBBACKUPTRIGGER

### Related Commands

Table 55 provides references to related commands.

*Table 55. Commands Related to DELETE DBBACKUPTRIGGER*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
QUERY DBBACKUPTRIGGER	Displays information about the database backup trigger.	Page 345
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514
UPDATE DBBACKUPTRIGGER	Changes one or more attributes of the database backup trigger.	Page 547

## DELETE DBVOLUME

---

### DELETE DBVOLUME (Delete a Database Volume)

Use the DELETE DBVOLUME command to delete an existing database volume or database volume copy. This command may require data on the volume to be moved to other database volumes.

*A group of mirrored volumes can consist of a database volume and up to two volume copies of that database volume. You cannot delete the last synchronized volume in a copy set unless the information on the volume can be moved by ADSM to other synchronized volumes in the database.*

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If a DELETE DBVOLUME background process is canceled, some of the database data may have already been moved. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶▶—DELeTe DBVolumE—volname————▶▶
```

### Parameters

*volname*

Specifies the name of the database volume to be deleted. This parameter is required.

### Examples

#### Task

Delete the database volume dvol1 located in d:\dbvols\

**Command:** delete dbvolume d:\dbvols\dvol1

#### Output:

```
ANR2243I Database volume d:\dbvols\dvol1 deleted.
```



## DELETE DBVOLUME

### Related Commands

Table 56 provides references to related commands.

*Table 56. Commands Related to DELETE DBVOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## DELETE DEVCLASS

---

### DELETE DEVCLASS (Delete a Device Class)

Use the DELETE DEVCLASS command to delete an existing device class.

In order to use this command, you must first delete all storage pools assigned to the specified device class and, if necessary, cancel any database dump, export, or import processes that are using the device class.

The server options file that is created during installation includes a DEVCONFIG option that specifies a default device configuration file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. The files you specify with this option are automatically updated with the results of this command. The files are also automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶▶—DELeTe DEVclAss—devclassname—◀◀
```

### Parameters

*devclassname*

Specifies the name of the device class to be deleted. This parameter is required.

### Examples

#### Task

Delete the device class named 8MMTAPE. There are no storage pools assigned to the device class.

**Command:** delete devclass 8mmtape

#### Output:

```
ANR2204I Device class 8MMTAPE deleted.
```

## DELETE DEVCLASS

### Related Commands

Table 57 provides references to related commands.

*Table 57. Commands Related to DELETE DEVCLASS*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
QUERY DEVCLASS	Displays information about one or more device classes.	Page 350
UPDATE DEVCLASS	Changes the attributes of a device class.	Page 550

## DELETE DOMAIN

---

### DELETE DOMAIN (Delete a Policy Domain)

Use the DELETE DOMAIN command to delete an existing policy domain. All assigned policy sets, management classes, and copy groups are deleted along with the specified policy domain. A policy domain with registered client nodes cannot be deleted.

#### Privilege Class

To issue this command, you must have system privilege.

#### Syntax

```
►►—DELeTe D0main—domainname—◄◄
```

#### Parameters

*domainname*

Specifies the name of the policy domain to be deleted. This parameter is required.

#### Usage Notes

You can delete the predefined STANDARD policy domain. If, however, at a later time you reinstall the ADSM server, all STANDARD policy objects are restored. Also, if after the server reinstall you are reestablishing your server database (via the database salvage commands), during DSMSEV AUDITDB command processing messages are issued indicating that an instance count does not agree with actual data. DSMSEV AUDITDB does not delete the STANDARD policy objects; however, it does synchronize the instance count to include the restored STANDARD policy objects.

#### Examples

##### Task

Delete the EMPLOYEE\_RECORDS policy domain.

**Command:** delete domain employee\_records

##### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR1501I Policy domain EMPLOYEE_RECORDS deleted.
```

## DELETE DOMAIN

### Related Commands

Table 58 provides references to related commands.

*Table 58. Commands Related to DELETE DOMAIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
COPY DOMAIN	Creates a copy of an existing policy domain.	Page 89
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
UPDATE DOMAIN	Changes the attributes of a policy domain.	Page 573

## DELETE DRIVE

---

### DELETE DRIVE (Delete a Drive from a Library)

Use the DELETE DRIVE command to delete a drive from a library. An active drive cannot be deleted.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶▶—DELEte Drive—libname—drivename—◀◀
```

### Parameters

*libname*

Specifies the name of the library from which the drive is to be deleted. This parameter is required.

*drivename*

Specifies the name of the drive to be deleted. This parameter is required.

### Examples

#### Task

Delete drive3 from the library named AUTO.

**Command:** delete drive auto drive3

#### Output:

```
ANR8412I Drive DRIVE3 deleted from library AUTO.
```

## DELETE DRIVE

### Related Commands

Table 59 provides references to related commands.

*Table 59. Commands Related to DELETE DRIVE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE LIBRARY	Deletes a library.	Page 220
QUERY DRIVE	Displays information about one or more drives.	Page 359
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
UPDATE DRIVE	Changes the attributes of a drive.	Page 575

## DELETE EVENT

---

### DELETE EVENT (Delete Event Records)

Use the DELETE EVENT command to delete event records from the database. An event record is created whenever processing of a scheduled command is started or missed.

Event records are deleted automatically based on the value set by the SET EVENTRETENTION command. While it is not necessary for you to manually delete event records, you can use this command to decrease the amount of database space required to hold event records, thus increasing the available capacity of the database.

An event record contains information regarding processing of scheduled work, such as the actual start date and start time, completion date and completion time, status, and results. This information is destroyed if you delete the event record. However, if you delete an event record, it does not alter defined schedules or the work which is pending for associated clients.

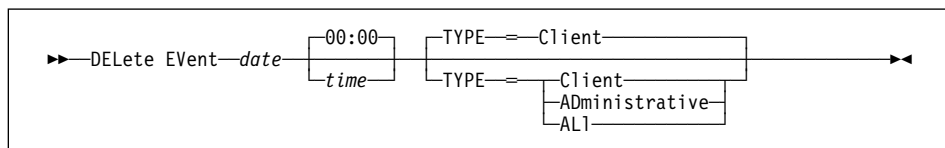
This command deletes event records if their scheduled start time is prior to a date and time specified by the administrator, unless the startup window for the event has not yet elapsed. If an event record has a startup window that has not elapsed, the record is not deleted.

This command only deletes the event records that exist at the time the command is executed.

### Privilege Class

To issue this command, you must have system privilege or unrestricted policy privilege.

### Syntax



### Parameters

*date*

Specifies the date that is used to determine which event records to delete. This parameter is required.

This parameter is used in conjunction with the TIME parameter to specify a date and time for deletion of event records. Any record whose scheduled start occurs prior to the specified date and time is deleted. However, records are not deleted for events whose startup window has not yet passed.



## DELETE EVENT

Possible values are:

### *MM/DD/YYYY*

Specifies that you want to delete records created on or before a specific date in MM/DD/YYYY format (for example, 01/31/1995).

### **TODAY**

Specifies that you want to delete records created on or before the current date.

### **TODAY**-*numdays* **or** -*numdays*

Specifies that you want to delete records created on or before the current date minus a specific number of days. The maximum value for the number of days is 9999.

For example, to delete records that are 30 or more days old, you can specify `TODATE=TODAY-30` or `TODATE=-30`.

### *time*

Specifies the time that is used to determine which event records to delete, in HH:MM format. This parameter is optional. The default value is 00:00.

This parameter is used in conjunction with the DATE parameter to specify a time and date for deletion of event records. Any record whose scheduled start occurs prior to the specified time and date is deleted. However, an exception is made for events whose startup window has not yet passed; records are not deleted for such events.

Possible values are:

### *HH:MM:SS*

Specifies that you want to delete records created on or before a specific time on the specified begin date using HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that you want to delete records created on or before the current time on the specified date.

### **NOW**+*numhours:numminutes* **or** +*numhours:numminutes*

Specifies that you want to delete records created on or before the current time plus a specific number of hours (and, optionally, minutes) on the specified date.

For example, if issue the DELETE EVENT command at 9:00 with `TOTIME=NOW+03:00` or `TOTIME=+03:00`, ADSM deletes records with a time of 12:00 or later on the specified date.

### **NOW**-*numhours:numminutes* **or** -*numhours:numminutes*

Specifies that you want to delete records created on or before the current time minus a specific number of hours (and, optionally, minutes) on the specified date.

## DELETE EVENT

For example, if issue the DELETE EVENT command at 9:00 with TOTIME=NOW-03:30 or TOTIME=-03:30, ADSM deletes records with a time of 5:30 or later on the specified date.

### **TYPE=***typevalue*

Specifies if events for client schedules, administrative command schedules, or both, are to be deleted. This parameter is optional. The default value is CLIENT. Possible values are:

#### **Client**

Specifies that events for client schedules are to be deleted.

#### **Administrative**

Specifies that events for administrative command schedules are to be deleted.

#### **ALI**

Specifies that events for both client and administrative command schedules are to be deleted.

## Examples

### **Task 1**

Delete records for events with scheduled start times prior to 00:00 on May 26, 1996 (05/26/1996), and whose startup window has passed. Records for these events are deleted regardless of whether the retention period for event records, as specified with the SET EVENTRETENTION command, has passed.

**Command:** DELEte EVent 05/26/1996

#### **Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR2512I Event records deleted: 18.
```

## DELETE EVENT

### Task 2

Delete records for events with scheduled start times prior to 08:00 on May 26, 1996 (05/26/1996), and whose startup window has passed. Records for these events are deleted regardless of whether the retention period for event records, as specified with the SET EVENTRETENTION command, has passed.

**Command:** DElete EVent 05/26/1996 08:00

#### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to following is displayed:

```
ANR2512I Event records deleted: 36.
```

### Related Commands

Table 60 provides references to related commands.

*Table 60. Commands Related to DELETE EVENT*

Command	Description	Reference
QUERY EVENT	Displays information about scheduled and completed events for selected clients.	Page 362
SET EVENTRETENTION	Specifies the number of days to retain event records.	Page 510

## DELETE FILESPACE

### DELETE FILESPACE (Delete Client Node Data from the Server)

Use the DELETE FILESPACE command to delete file spaces from the server. Files that belong to the specified file space are deleted from primary and copy storage pools. Remember that file space names are case sensitive. To determine the correct capitalization for the file space to be deleted, use the QUERY FILESPACE command.

Use the DELETE FILESPACE command to:

- Remove files for users who are not authorized to delete backed up or archived files from storage pools.
- Remove a client node from ADSM. Before removing a client node, you must delete all files belonging to the client node.
- Delete files belonging to a specific owner. When you delete a file space belonging to a specific owner, only the files that have the specified owner name in the file space are deleted.

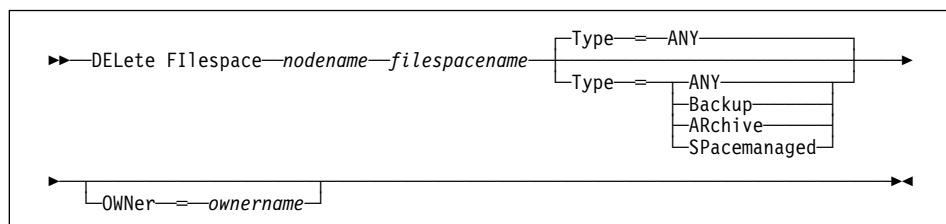
The DELETE FILESPACE command creates a server process that deletes one or more file spaces as a series of batch database transactions, thus preventing a rollback or commit for an entire file space as a single action. If the DELETE FILESPACE process is canceled or if a system failure occurs, a partial deletion can occur. In this case, a subsequent DELETE FILESPACE command for the same node or owner can delete the remaining data.

This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command. If a DELETE FILESPACE background process is canceled, some or all of the client's file spaces are deleted. If a partial deletion occurred, use the QUERY FILESPACE command to determine which file spaces have not been deleted. From the client node, the remaining files in the file space can be viewed with commands such as QUERY BACKUP. For information about the query commands, refer to the user's guide for your client and platform.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.

### Syntax



## DELETE FILESPACE

### Parameters

*nodename*

Specifies the name of the client node to which the file space belongs. This parameter is required.

*filespace*

Specifies the name of the file space to be deleted. This name is case-sensitive. This parameter is required. You can use a pattern-matching expression to specify this name. All matching file space names are included.

**Type**=*datatype*

Specifies the type of data to be deleted. This parameter is optional. The default value is ANY. Possible values are:

**ANY**

Specifies to delete, in order, only backed-up versions of files and archived copies of files.

To delete space-managed files (files migrated from HSM client nodes), you must specify TYPE=SPACEMANAGED.

If you specify DELETE FILESPACE *nodename* \* TYPE=ANY, all backed-up and archived files in all file spaces for that node will be deleted. File spaces will only be deleted if they do not contain files migrated from an HSM client.

**Backup**

Specifies to delete all backup data.

**AR**chive

Specifies to delete all archive data.

**SP**acemanaged

Specifies to delete files migrated from a user's local file system (space-managed files from an HSM client). The OWNER parameter is ignored when you specify TYPE=SPACEMANAGED.

**OWN**er=*ownername*

Specifies to restrict the data that is deleted from a file space to files belonging to the owner named. This parameter is optional and is ignored when TYPE=SPACEMANAGED. This parameter only applies to multiuser client systems such as AIX.

## DELETE FILESPACE

### Examples

#### Task 1

Delete all backup and archive data in file spaces that belong to the client node CLIENT1.

**Command:** DElete Filespace client1 \*

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANS5104I Process number 2 started.
```

#### Task 2

Delete the file space named C\_Drive that belongs to the client node HTANG.

**Command:** DElete Filespace htang C\_Drive

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANS5104I Process number 3 started.
```

#### Task 3

Delete all files migrated from client node APOLLO (that is, all space-managed files).

**Command:** DElete Filespace apollo \* Type=SPacemanaged

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

## DELETE FILESPACE

```
ANR0824I Delete Filespace * (space-managed data) for node
APOLLO started.
ANR0800I DELETE FILESPACE: * for node APOLLO started as
process 4.
adsm>
ANR0806I Delete Filespace * complete for node APOLLO:
8 files deleted.
```

### Related Commands

Table 61 provides references to related commands.

*Table 61. Commands Related to DELETE FILESPACE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY FILESPACE	Displays information about data in one or more filespaces that belong to a client.	Page 377
QUERY OCCUPANCY	Displays file space information by storage pool.	Page 408
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
REMOVE NODE	Removes a client from the list of registered nodes for a specific policy domain.	Page 472
RENAME FILESPACE	Renames a client filespace on the server.	Page 476

## DELETE LIBRARY

---

### DELETE LIBRARY (Delete a Library)

Use the DELETE LIBRARY command to delete an existing library. Prior to deleting a library, all assigned drives must be deleted.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶▶—DELeTe LIBRary—libraryname—————▶▶
```

### Parameters

*libraryname*

Specifies the name of the library to be deleted. This parameter is required.

### Examples

#### Task

Delete the MANLIB library.

**Command:** DELeTe LIBRary manlib

#### Output:

```
ANR8410I Library MANLIB deleted.
```



## DELETE LIBRARY

### Related Commands

Table 62 provides references to related commands.

*Table 62. Commands Related to DELETE LIBRARY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE DRIVE	Deletes a drive from a library.	Page 210
QUERY DRIVE	Displays information about one or more drives.	Page 359
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
UPDATE DRIVE	Changes the attributes of a drive.	Page 575
UPDATE LIBRARY	Changes the attributes of a library.	Page 577

## DELETE LOGVOLUME

---

### DELETE LOGVOLUME (Delete a Recovery Log Volume)

Use the DELETE LOGVOLUME command to delete an existing recovery log volume or recovery log volume copy. This command may require data on the volume to be moved to other database volumes.

A *group of mirrored volumes* can consist of a recovery log volume and up to two volume copies of that recovery log volume. You cannot delete the last synchronized volume in a copy set unless the information on the volume can be moved by ADSM to other synchronized volumes in the recovery log.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

#### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

#### Syntax

```
▶▶—DELeTe LOGVolumE—volname————▶▶
```

#### Parameters

*volname*

Specifies the name of the recovery log volume to be deleted. This parameter is required.

#### Examples

##### Task

Delete recovery log volume reclog located in d:\rlogvol\ . No data resides on the volume.

**Command:** delete logvolume d:\rlogvol\reclog

##### Output:

```
ANR2264I Delete process initiated for recovery
log volume d:\rlogvol\reclog (process id 203).
ANR0245I Deletion complete for recovery log volume
d:\rlogvol\reclog.
```

## DELETE LOGVOLUME

### Related Commands

Table 63 provides references to related commands.

*Table 63. Commands Related to DELETE LOGVOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
QUERY LOGVOLUME	Displays information about one or more log volumes.	Page 394
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## DELETE MGMTCLASS

---

### DELETE MGMTCLASS (Delete a Management Class)

Use the DELETE MGMTCLASS command to delete an existing management class, including the assigned default management class. You cannot delete a management class in the ACTIVE policy set. All assigned copy groups are deleted along with the specified management class. If you are deleting the default management class, remember a policy set cannot be activated unless it has an assigned default management class.

#### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the management class belongs.

#### Syntax

```
▶▶—DELeTe MGmtclass—domainname—setname—classname————▶▶
```

#### Parameters

##### *domainname*

Specifies the name of the policy domain to which the management class belongs. This parameter is required.

##### *setname*

Specifies the name of the policy set to which the management class belongs. This parameter is required.

##### *classname*

Specifies the name of the management class that you want to delete. This parameter is required.

#### Usage Notes

You can delete the predefined STANDARD management class located in the STANDARD policy domain. If, however, at a later time you reinstall the ADSM server, all STANDARD policy objects are restored. Also, if after the server reinstall you are reestablishing your server database (via the database salvage commands), during DSMSERV AUDITDB command processing messages are issued indicating that an instance count does not agree with actual data. DSMSERV AUDITDB does not delete the STANDARD policy objects; however, it does synchronize the instance count to include the restored STANDARD policy objects.

#### Examples

##### Task

Delete the ACTIVEFILES management class from the VACATION policy set of the EMPLOYEE\_RECORDS policy domain.

**Command:** DELeTe MGmtclass employee\_records vacation activefiles

## DELETE MGMTCLASS

### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR1521I Management class ACTIVEFILES deleted  
from policy domain EMPLOYEE_RECORDS, set VACATION.
```

### Related Commands

Table 64 provides references to related commands.

*Table 64. Commands Related to DELETE MGMTCLASS*

Command	Description	Reference
ASSIGN DEFMGMTCLASS	Assigns an existing management class as the default for a specified policy set.	Page 48
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581

## DELETE POLICYSET

---

### DELETE POLICYSET (Delete a Policy Set)

Use the DELETE POLICYSET command to delete an existing policy set. You cannot delete the ACTIVE policy set. When a policy set is deleted all management classes and copy groups that belong to the policy set within the specified policy domain are deleted.

#### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

#### Syntax

```
►►—DELEte POLIcYset—domainname—setname—————►◄
```

#### Parameters

*domainname*

Specifies the name of the policy domain to which the policy set belongs. This parameter is required.

*setname*

Specifies the name of the policy set to be deleted. This parameter is required.

#### Usage Notes

You can delete the predefined STANDARD policy set. If, however, at a later time you reinstall the ADSM server, all STANDARD policy objects are restored. Also, if after the server reinstall you are reestablishing your server database (via the database salvage commands), during DSMSEV AUDITDB command processing messages are issued indicating that an instance count does not agree with actual data. DSMSEV AUDITDB does not delete the STANDARD policy objects; however, it does synchronize the instance count to include the restored STANDARD policy objects.

## DELETE POLICYSET

### Examples

#### Task

Delete the VACATION policy set from the EMPLOYEE\_RECORDS policy domain.

**Command:** DELEte P0licyset employee\_records vacation

#### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR1511I Policy set VACATION deleted from policy domain  
EMPLOYEE_RECORDS.
```

## DELETE POLICYSET

### Related Commands

Table 65 provides references to related commands.

*Table 65. Commands Related to DELETE POLICYSET*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY POLICYSET	Creates a copy of an existing policy set.	Page 94
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625



## DELETE SCHEDULE

---

### DELETE SCHEDULE (Delete a Client or an Administrative Command Schedule)

The DELETE SCHEDULE command takes two forms, depending on whether the schedule applies to client operations or administrative commands. The syntax and parameters for each form are defined separately.

---

#### Deleting Client Schedules

Use the DELETE SCHEDULE command to remove one or more existing schedules from the database.

Any client associations to a schedule are removed when the schedule is deleted.

#### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the specified policy domain.

#### Syntax

```
►►—DELeTe SChedule—domainname—schedulename—[Type=Client]—►►
```

#### Parameters

*domainname*

Specifies the name of the policy domain to which the schedule belongs. This parameter is required.

*schedulename*

Specifies the name of the schedule to be deleted. This parameter is required. You can use a pattern-matching expression to specify this name. All matching schedule names are deleted from the specified policy domain.

**Type=Client**

Specifies that this command should delete a client schedule. This parameter is optional. The default value is CLIENT. Administrative command schedules are deleted by specifying TYPE=ADMINISTRATIVE.

---

#### Deleting Administrative Command Schedules

Use the DELETE SCHEDULE command to remove one or more existing schedules from the database.

#### Privilege Class

You must have system authority to delete an administrative command schedule.

## DELETE SCHEDULE

### Syntax for Deleting Administrative Command Schedules

```
►►—DELeTe SChedule—schedulename—Type—=—Administrative—►►
```

### Parameters for Deleting Administrative Command Schedules

*schedulename*

Specifies the name of the schedules to be deleted. This parameter is required. You can use a pattern-matching expression to specify this name.

**Type=Administrative**

Specifies that the schedule to be deleted is an administrative schedule. This parameter is required for deleting administrative command schedules.

### Examples

#### Task 1

Delete the WEEKLY\_BACKUP schedule, which belongs to the EMPLOYEE\_RECORDS policy domain.

**Command:** `delete schedule employee_records weekly_backup`

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the enter key. A message similar to the following is displayed:

```
ANR2501I Schedule WEEKLY_BACKUP deleted from policy domain  
EMPLOYEE_RECORDS.
```

#### Task 2

Delete all schedules that belong to the policy domain DOMAIN1.

**Command:** `delete schedule domain1 *`

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the Enter key. A message similar to the following is displayed:

## DELETE SCHEDULE

```
ANR2501I Schedule WINTER deleted from policy domain DOMAIN1.  
ANR2501I Schedule SUMMER deleted from policy domain DOMAIN1.  
ANR2501I Schedule SPRING deleted from policy domain DOMAIN1.
```

### Task 3

Delete administrative schedule named WEEKLY from the database.

**Command:** delete schedule weekly type=administrative

**Output:**

```
Do you wish to proceed? (Y/N)
```

Type Y and press the Enter key. A message similar to the following is displayed:

```
ANR2504I Schedule WEEKLY deleted.
```

### Related Commands

Table 66 provides references to related commands.

*Table 66. Commands Related to DELETE SCHEDULE*

Command	Description	Reference
COPY SCHEDULE	Creates a copy of an existing schedule.	Page 96
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164
QUERY SCHEDULE	Displays information about one or more schedules.	Page 422
UPDATE SCHEDULE	Changes the attributes of a schedule.	Page 590

## DELETE STGPOOL

---

### DELETE STGPOOL (Delete a Storage Pool)

Use the DELETE STGPOOL command to delete a storage pool.

In order to use this command, you must first delete all volumes assigned to the specified storage pool.

You cannot delete a storage pool that is defined as a subordinate storage pool. For more information on subordinate storage pools, see the NEXTSTGPOOL parameter in the DEFINE STGPOOL command.

Do not delete a storage pool that is specified as a destination for a management class or copy group in the ACTIVE policy set.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
▶▶—DELeTe STGpool—poolname————▶▶
```

### Parameters

*poolname*

Specifies the name of the storage pool to be deleted. This parameter is required.

### Examples

#### Task

Delete the storage pool named POOLA.

**Command:** DELeTe STGpool poola

#### Output:

```
ANR2201I Storage pool POOLA deleted.
```

## DELETE STGPOOL

### Related Commands

Table 67 provides references to related commands.

*Table 67. Commands Related to DELETE STGPOOL*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
DEFINE STGPOOL	Defines a storage pool as a named collection of data storage media.	Page 180
QUERY STGPOOL	Displays information about one or more storage pools.	Page 436
UPDATE STGPOOL	Changes the attributes of a storage pool.	Page 605

## DELETE VOLHISTORY

---

### DELETE VOLHISTORY (Delete Sequential Volume History Information Collected by the Server)

Use the DELETE VOLHISTORY command to delete sequential volume history information collected by the server when the information is no longer needed. For example, you may want to delete information about volumes used for older, obsolete database backups.

Volume history information includes data such as date and time of use for the following types of volumes:

- Volumes not in storage pools
  - Volumes used to back up an ADSM database (full or incremental)
  - Volumes used to dump an ADSM database
  - Volumes used to export administrator, node, policy, or server data
- Volumes in storage pools
  - Storage pool sequential access volumes that have been added
  - Storage pool sequential access volumes that have been reused in storage pools through reclamation or MOVE DATA operations
  - Storage pool sequential access volumes that have been removed by using the DELETE VOLUME command or removed during reclamation of scratch volumes

When volume history information about volumes *not* in storage pools is deleted, the volumes return to scratch status if they were acquired by ADSM as scratch volumes. For scratch volumes with device type FILE, the files are deleted.

When volume history information about volumes in storage pools is deleted, the volumes themselves are not affected and remain in the ADSM database.

To display sequential volume history information collected by the server, use the QUERY VOLHISTORY command.

To save sequential volume history information collected by the server to one or more files, use the BACKUP VOLHISTORY command.

To delete records from volume history information on a regular basis, use the DEFINE SCHEDULE command.

## DELETE VOLHISTORY

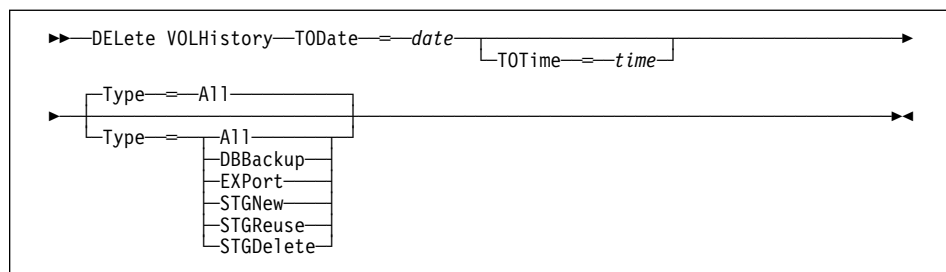
### Notes:

1. Existing volume history files are *not* automatically updated with this command.
2. Do not delete sequential volume history information until you are sure that you no longer need it. For example, do not delete dump volume information or storage volume reuse information unless you have backed up or dumped the database at a later time than that specified for the delete operation.
3. Do not delete the volume history information for database dump, database backup, or export volumes that reside in automated libraries unless you want to return the volumes to scratch status. When the DELETE VOLHISTORY command removes volume information for database dump, database backup, or export volumes, the volumes are automatically returned to scratch status if they reside in automated libraries. The result is that these volumes are available for reuse by the server and the information stored on them may be overwritten when the server reuses the volume for some other purpose, such as storage pool volumes or other database backups.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### TODate=*date*

Specifies the date to use to select sequential volume history information to be deleted. ADSM deletes only those records with a date the same as or older than the date you specify.

This parameter is required. Possible values are:

#### MM/DD/YYYY

Specifies that you want to delete records created on or before a specific date in MM/DD/YYYY format (for example, 01/31/1995).

#### TODAY

Specifies that you want to delete records created on or before the current date.

## DELETE VOLHISTORY

### **TODAY**–*numdays* or –*numdays*

Specifies that you want to delete records created on or before the current date minus a specific number of days. The maximum value for the number of days is 9999.

For example, to delete records that are 30 or more days old, you can specify `TODATE=TODAY-30` or `TODATE=-30`.

### **TOTime**=*time*

Specifies the time to use to select records to delete from the volume history file. This parameter is optional.

Of the records that match the date criteria you specify with the `TODATE` parameter, ADSM deletes only those records with a time that is the same as or earlier than the time you specify with the `TOTIME` parameter. This parameter is optional. The default value is midnight (00:00:00) on the specified date. Possible values are:

### *HH:MM:SS*

Specifies that you want to delete records created on or before a specific time on the specified date in HH:MM:SS format (for example, 12:30:22).

### **NOW**

Specifies that you want to delete records created on or before the current time on the specified date.

For example, if you issue the `DELETE VOLHISTORY` command at 9:00 with `TOTIME=NOW`, ADSM deletes records with a time of 9:00 or earlier on the specified date.

### **NOW**+*numhours:numminutes* or +*numhours:numminutes*

Specifies that you want to delete records created on or before the current time plus a specific number of hours (and optionally, minutes) on the specified date.

For example, if you issue the `DELETE VOLHISTORY` command at 9:00 with `TOTIME=NOW+3` or `TOTIME=+3`, ADSM deletes records with a time of 12:00 or earlier on the specified date.

### **NOW**–*numhours:numminutes* or –*numhours:numminutes*

Indicates that you want to delete records created on or before the current time minus a specific number of hours (and optionally, minutes) on the specified date.

For example, if you issue the `DELETE VOLHISTORY` command at 9:00 with `TOTIME=NOW-3:30` or `TOTIME=-3:30`, ADSM deletes records with a time of 5:30 or earlier on the specified date.

### **Type**=*typevalue*

Specifies the type of records to delete from the volume history file. This parameter is optional. The default value is ALL. Possible values are:



## DELETE VOLHISTORY

### All

Specifies to delete all records that meet the date and time criteria you specify.

**Note:** If you previously had an ADSM Version 1 Release 1 or 2 server installed, and you used the DUMP DB command to create online database dumps, records related to the volumes used for those dumps can exist in your volume history information. When you specify TYPE=ALL for the DELETE VOLHISTORY command, ADSM deletes any of those records that meet the date and time criteria you specify.

### DBBackup

Specifies to delete only records that both contain information about volumes used for backups (full or incremental), and meet the date and time criteria you specify.

### EXPort

Specifies to delete only records that both contain information about volumes used for exporting data, and meet the date and time criteria you specify.

### STGNew

Specifies to delete only records that both contain information about new sequential access storage volumes, and meet the date and time criteria you specify.

### STGReuse

Specifies to delete only records that both contain information about reused sequential storage pool volumes, and meet the date and time criteria you specify.

### STGDelete

Specifies to delete only records that both contain information about deleted sequential storage volumes, and meet the date and time criteria you specify.

## DELETE VOLHISTORY

### Examples

#### Task

Delete all volume history information created on or before 03/28/1996.

**Command:** DElete VOLHistory T0Date=03/28/1996 Type=All

#### Output:

```
Do you wish to proceed? (Yes/No)
```

Type Y and press the Enter key. A message similar to the following is displayed.

```
ANR2467I DELETE VOLHISTORY: 1 sequential volume history  
entries were successfully deleted.
```

## DELETE VOLHISTORY

### Related Commands

Table 68 provides references to related commands.

*Table 68. Commands Related to DELETE VOLHISTORY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP VOLHISTORY	Records sequential volume history information in external files.	Page 72
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
QUERY VOLHISTORY	Displays sequential volume history information that has been collected by the server.	Page 443

## DELETE VOLUME

---

### DELETE VOLUME (Delete a Storage Pool Volume)

Use the DELETE VOLUME command to delete a storage pool volume and, optionally, the files within the volume. This command can be used to delete a volume assigned to either a primary or copy storage pool. If during the processing of this command for a primary storage pool volume, ADISM deletes the primary copy of a file (not a cached copy), ADISM also deletes any copies of that file residing in copy storage pools.

If you are deleting multiple volumes, it is recommended that you delete the volumes one at a time. Concurrently deleting multiple volumes can adversely affect server performance.

To determine the contents stored on a volume, use the QUERY CONTENT command.

Before you can delete a storage volume, you must do one of the following:

- Use the MOVE DATA command to move all files to another volume.
- Explicitly request to discard all files in the storage volume by specifying DISCARDATA=YES.

Storage pool volumes may not be totally empty after a move data operation completes if one or more files cannot be relocated to another volume due to input/output errors on the device or if integrity errors were found in the file's structure. In this case, a DELETE operation with DISCARDATA=YES is required to fully delete the volume. The files with I/O or integrity errors are then lost.

If you specify DISCARDATA=YES, ADISM creates a server process that deletes files as a series of batch database transactions. If the DELETE VOLUME operation is canceled during processing or if a system failure occurs, a partial deletion can occur. In this case, a subsequent DELETE VOLUME command for the same volume continues deleting the remaining files.

Storage pool volumes cannot be deleted if they are already in use. For example, volumes cannot be deleted if a user is restoring or retrieving a file residing in the storage volume, if ADISM is writing information to the volume, or if reclamation is in process.

This command generates a server process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

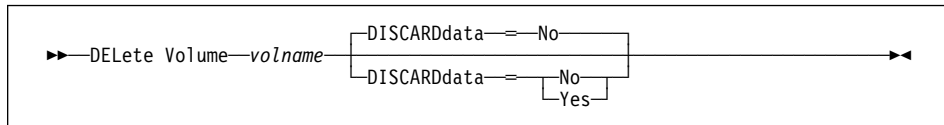
Do not use the DELETE VOLUME command with DISCARDATA=YES if a restore process (RESTORE STGPOOL or RESTORE VOLUME) is running. The DELETE VOLUME command could cause the restore to be incomplete.

### Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool to which the volume is defined.

## DELETE VOLUME

### Syntax



### Parameters

*volname*

Specifies the name of the volume to be deleted. This parameter is required.

**DISCARDdata**=*discardvalue*

Specifies whether all files stored in the specified volume are deleted from the storage pool. This parameter is optional. No sequential access media mounts are required for this type of deletion. The default value is NO. Possible values are:

**No**

Specifies that all files stored in the specified volume are not deleted.

**Yes**

Specifies that all files stored in the specified volume are deleted.

When this option is specified for a primary storage pool volume, the command deletes from the volume both primary copies of files and files that are cached copies (copies left behind by migration processes). When primary copies of files are deleted from the volume, any copies of these files in copy storage pools are also deleted.

### Examples

#### Task

Delete storage pool volume spvol located in d:\admserv\.

**Command:** delete volume d:\admserv\spvol

#### Output:

```
ANR2220W This command will delete volume
d:\admserv\spvol from its storage pool after
verifying that the volume contains no data.
Do you wish to proceed? (Yes/No)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR2208I Volume d:\admserv\spvol deleted from
storage pool FILEPOOL.
```

## DELETE VOLUME

### Related Commands

Table 69 provides references to related commands.

*Table 69. Commands Related to DELETE VOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
MOVE DATA	Moves data from a specified storage pool volume to another storage pool volume.	Page 309
QUERY CONTENT	Displays information about one or more files currently residing in a storage pool volume.	Page 328
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY VOLUME	Displays information about one or more storage pool volumes.	Page 450
UPDATE VOLUME	Updates the attributes of one or more storage pool volumes.	Page 619

## DISABLE

---

### DISABLE (Temporarily Prevent Client Node Access to the Server)

Use the DISABLE command to prevent client nodes from starting any new sessions. Current client node sessions are allowed to complete, but no new client node sessions are allowed to start.

This command does not affect system processes (like migration and reclamation).

All registered administrators can access the server when it is disabled with the DISABLE command. To allow client nodes to access the server, use the ENABLE command.

Use the QUERY STATUS command to display the current availability of the server. The DISABLE command can be used before canceling all client node sessions with the CANCEL SESSION command so that the clients cannot automatically reconnect.

### Privilege Class

To issue this command, you must have system privilege or operator privilege.

### Syntax

```
▶▶—DISAbLe—▶▶
```

### Parameters

None.

### Examples

#### Task

Temporarily prevent new client node sessions from accessing the server.

**Command:** disable

#### Output:

```
ANR2097I Server now disabled for backup/archive client  
access.
```

## DISABLE

### Related Commands

Table 70 provides references to related commands.

*Table 70. Commands Related to DISABLE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL SESSION	Cancels one or all active sessions with the server.	Page 78
ENABLE	Resumes server activity following the DISABLE command.	Page 246
QUERY SESSION	Displays information about all active administrator and client sessions with ADSM.	Page 427
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431



## DISMOUNT VOLUME

---

### DISMOUNT VOLUME (Dismount a Volume by Volume Name)

Use the DISMOUNT VOLUME command to dismount a volume by volume name.

You can only use this command to dismount *idle* volumes. Use the QUERY MOUNT command to determine the status of any mounted volumes.

#### Privilege Class

To issue this command, you must have system privilege or operator privilege.

#### Syntax

```
▶▶—DISMount Volume—volName————▶▶
```

#### Parameters

*volname*

Specifies the name of the volume to be dismounted. This parameter is required.

#### Examples

##### Task

Dismount the volume BTV005.

**Command:** `dismount volume btv005`

##### Output:

```
ANR5280I Command Accepted.
```

#### Related Commands

Table 71 provides references to related commands.

---

*Table 71. Commands Related to DISMOUNT VOLUME*

Command	Description	Reference
QUERY MOUNT	Displays information about mounted sequential access media.	Page 401

---

## ENABLE

---

### ENABLE (Resume User Activity on the Server)

Use the ENABLE command to permit client node activity to begin after issuing the DISABLE command.

This command does not affect system processes (like migration and reclamation).

Use the QUERY STATUS command to display the current availability of the server.

### Privilege Class

To issue this command, you must have system privilege or operator privilege.

### Syntax

```
▶▶—ENable—————▶▶
```

### Parameters

None.

### Examples

#### Task

Resume normal operation, permitting client nodes to access the server.

**Command:** enable

**Output:**

```
ANR2096I Server now enabled for backup/archive access.
```

## ENABLE

### Related Commands

Table 72 provides references to related commands.

*Table 72. Commands Related to ENABLE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL SESSION	Cancels one or all active sessions with the server.	Page 78
DISABLE	Prevents new clients from accessing ADSM but permits existing sessions to continue.	Page 243
QUERY SESSION	Displays information about all active administrator and client sessions with ADSM.	Page 427
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## EXPIRE INVENTORY

---

### EXPIRE INVENTORY (Manually Start Inventory Expiration Processing)

Use the EXPIRE INVENTORY command to manually start inventory expiration processing. Only one expiration process is allowed at any time; therefore, if expiration processing is currently running, another process cannot be started. Expiration eligibility is determined by policy information assigned to either the backup or archive files. If a file is eligible for expiration, the storage used by this file is deallocated.

Automatic expiration processing is controlled by the EXPINTERVAL option specified in the ADSM server options file. If EXPINTERVAL is set to 0, no automatic expiration will occur, and this command will be necessary to start expiration processing. For additional information about the EXPINTERVAL parameter, see Appendix E, "Tailoring Server Options" on page 677.

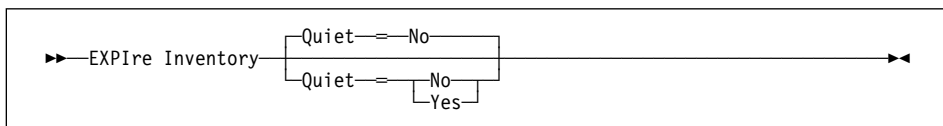
This command creates a background process that can be canceled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

The inventory expiration process removes client backup and archive file copies from server storage under the terms specified by the management classes to which the files are bound.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### Quiet<sub>quietvalue</sub>

Specifies whether to suppress detailed messages during ADSM's expiration processing. Normally ADSM sends detailed messages when you make policy changes that affect client files, such as deleting a management class or copy group. You can use this option to reduce the number of messages about policy changes that are generated during expiration processing.

When you specify QUIET=YES, ADSM will issue messages about policy changes only when files are deleted and either the default management class or retention grace period for the domain has been used to expire the files. This parameter is optional.

You can also specify the EXPQUIet option in the server options file to automatically determine if expiration processing is performed with minimal summary messages.

## EXPIRE INVENTORY

The values are:

**No**

Specifies that ADSM sends detailed informational messages. No is the default.

**Yes**

Specifies that ADSM sends only minimal summary messages and only for files that have expired based on the copy group in the default management class or retention grace period for the domain.

### Examples

**Task**

Issue the EXPIRE INVENTORY command.

**Command:** expire inventory

**Output:**

```
ANR0811I Inventory client file expiration started as
process 2.
ANR0812I Inventory file expiration process 2 completed:
deleted 25 backup files and 3 archive files.
```

### Related Commands

Table 73 provides references to related commands.

*Table 73. Commands Related to EXPIRE INVENTORY*

Command	Description	Reference
AUDIT LICENSES	Checks for compliance with defined licenses.	Page 54
CANCEL PROCESS	Cancels a background server process.	Page 74
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## EXPORT ADMIN

### EXPORT ADMIN (Copy Administrator Information to Sequential Media)

Use the EXPORT ADMIN command to export administrator and authority definitions for one or more administrators from the ADSM server to sequential media.

The EXPORT ADMIN command exports administrator information such as:

- Administrator name, password, and contact information
- Administrative privilege classes the administrator has been granted

The EXPORT ADMIN command allows administrative information to be copied to another server. In order to copy ADSM data and then restore it to a new server, the device class must be supported on both server platforms.

This command is useful for permitting later retrieval of server information for copying the information a new server.

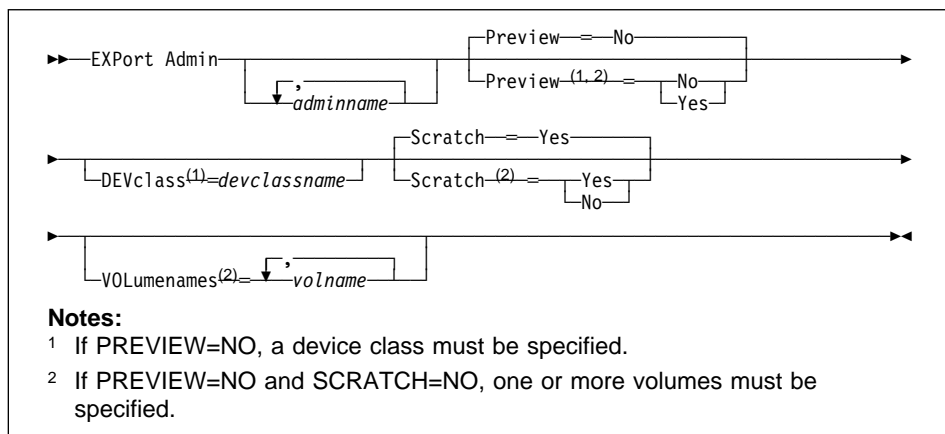
You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the export operation. You can also view this information from the server console.

This command generates a background process that can be canceled with the CANCEL PROCESS command. If an EXPORT ADMIN background process is canceled, the sequential media holding the exported data are incomplete and should not be used for importing data. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *adminname*

Specifies a list of administrators for which administrator information is to be exported. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use pattern-matching expressions to specify names. All matching administrators are included in the list. If you do not specify a value for this parameter, information related to all administrators is exported.

#### **Preview**=*value*

Specifies whether you want to preview the results of the export operation, without actually exporting information. You can use this parameter to preview how many bytes of data are transferred, allowing you to determine how many volumes will be required to hold the data. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the administrator information is to be exported. If you specify this value, you must specify a value for the DEVCLASS parameter.

#### **Yes**

Specifies that you want to preview the results of the export operation, without actually exporting administrator information. Information on the number and types of objects exported, together with the number of bytes transferred are reported to the server console and the activity log. If you specify this value, you do not need to specify device or volume parameters.

#### **DEVclass**=*devclassname*

Specifies the device class to which the export data is to be written. This parameter is required unless you specified PREVIEW=YES. You cannot specify the DISK device class.

If the mount limit for the device class specified has already been reached (all drives are busy) when the export runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.

#### **Scratch**=*scratchvalue*

Specifies whether scratch volumes can be used for export. Scratch volumes are not supported for LAN3995 device classes. The default is SCRATCH=YES. Possible values are:

#### **Yes**

Specifies that scratch volumes are to be used for export. If you include a list of volumes on the VOLUMENAMES parameter, then scratch volumes are used only if there is not enough space to store the exported files on the volumes specified. If the device type associated with the specified device class is FILE, a file named 00000001.EXP will be chosen for the first scratch volume. If a file already exists by this name in the target directory, you must first rename or erase it if you want to use SCRATCH=YES.

## EXPORT ADMIN

### No

Specifies that scratch volumes cannot be used for export. You must include a list of volumes on the VOLUMENAMES parameter to contain all of the export data. To determine how many volumes you may need, you can run the command specifying PREVIEW=YES.

### **VOL**umenames=*volname*

Specifies one or more sequential volumes that are to be used to contain exported files. This parameter is optional. The items in the list are separated by commas, with no intervening spaces.

A volume name must be specified if SCRATCH=NO and PREVIEW=NO. If you do not specify a volume and SCRATCH=YES, scratch volumes are used. Any volume name used by ADSM cannot contain any embedded blanks or equal signs.

Use the following naming conventions to specify volume names for each device type:

### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

### **4MM, 8MM, DLT, QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.

### **OPTICAL or MO35**

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### **LAN3995**

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

## Examples

### Task

Export the administrator information for all defined administrators to optical storage (assuming a device class named SQUARE1).

The number and type of objects exported are reported to the system console and in the activity log.

**Command:** export admin devclass=square1 Scratch=Yes

**Output:**



## EXPORT ADMIN

```
ANR0609I EXPORT ADMIN started as process 9.  
ANS5104I Process number 9 started.
```

### Related Commands

Table 74 provides references to related commands.

*Table 74. Commands Related to EXPORT ADMIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT NODE	Copies client node information to external media.	Page 254
EXPORT POLICY	Copies policy information to external media.	Page 260
EXPORT SERVER	Copies all or part of the server to external media.	Page 264
IMPORT ADMIN	Restores administrative information from external media.	Page 283
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## EXPORT NODE

---

### EXPORT NODE (Copy Client Node Information to Sequential Media)

Use the EXPORT NODE command to export client node definitions from the ADSM server to sequential media. Each client node definition includes:

- User ID, password, and contact information
- Name of the policy domain to which the client is assigned
- File compression status
- Whether the user has the authority to delete backed up or archived files from data storage
- Whether the client node ID was locked at the time of export

Optionally, the EXPORT NODE command exports node data information such as:

- File space definitions
- Data from files migrated from HSM client nodes
- Archive or backup file copy data
- Access authorization information pertaining to the file spaces exported.

The EXPORT NODE command allows client node information to be copied to another server. In order to back up ADSM data and then restore it to a new server, the sequential device must be supported on both platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

This command is useful for permitting later retrieval of server information for copying the information to a new server.

If a primary file cannot be exported because it is damaged or because it resides on a volume that is UNAVAILABLE or OFFLINE, ADSM attempts to obtain a copy of the file from a copy storage pool.

You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the export operation.

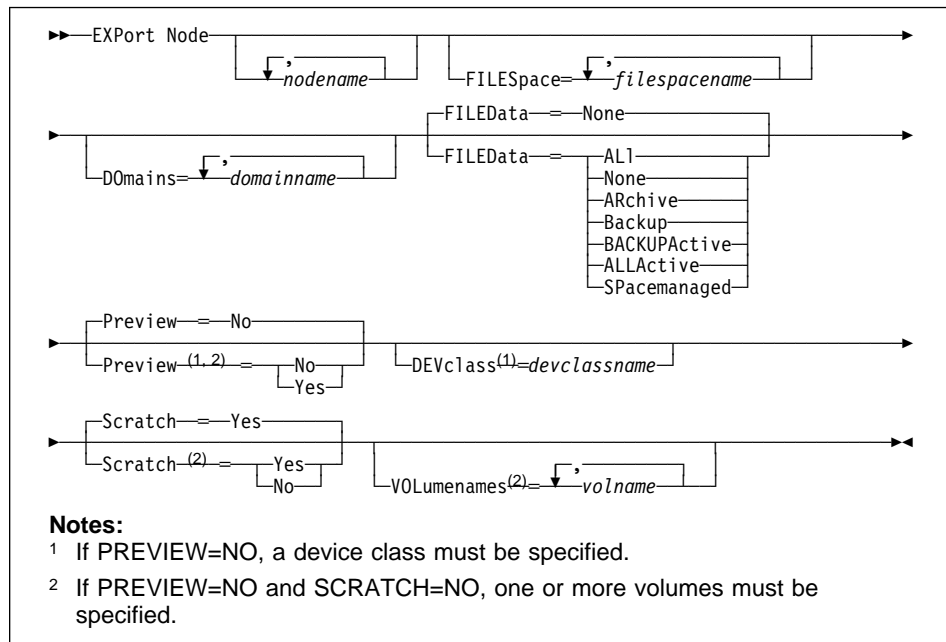
This command generates a background process that can be canceled with the CANCEL PROCESS command. If an EXPORT NODE background process is canceled, the sequential media holding the exported data are incomplete and should not be used for importing data. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

## EXPORT NODE

### Syntax



### Parameters

#### *nodename*

Specifies a list of client node names for which information is to be exported. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use pattern-matching expressions to specify names. All matching nodes are included in the list. If you do not specify a value for this parameter, information related to all nodes is exported.

#### **FILESpace=filespaceName**

Specifies a list of file space names for which data is to be exported. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use a pattern-matching expression to specify a name. All matching names are included in the list. If you do not specify a value for this parameter, information related to all files space names is exported.

#### **DOMains=domainName**

Specifies a list of policy domain names from which nodes should be exported. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. For a node to be exported it must belong to one of the specified domains. You can use a pattern-matching expression to specify a name. All matching domains are included in the list. If you do not specify a value for this parameter, nodes assigned to all domains is exported.

## EXPORT NODE

### **FILEData=***filedatatype*

Specifies the type of files that should be exported for all nodes defined to the server. This parameter is optional. The default value is NONE.

The device class used by the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, then two drives are needed to export node information. The mount limit value for the device class needs to be set to at least 2.

The following descriptions mention *active* and *inactive* backup file copies. An active backup file copy is the most recent backup copy for a file that still exists on the client workstation. All other backup file copies are called inactive copies. Possible values are:

#### **ALI**

Specifies that ADSM exports all backup versions of files, all archived files, and all files migrated from a user's local file system.

#### **None**

Specifies that no files are to be exported; only node definitions are to be exported.

#### **ARChive**

Specifies that only archived files are to be exported.

#### **Backup**

Specifies that only backup versions, whether active or inactive, are to be exported.

#### **BACKUPActive**

Specifies that only active backup versions are to be exported.

#### **ALLActive**

Specifies that ADSM exports all active backup versions of files, all archived files, and all files migrated from a user's local file system.

#### **SPacemanaged**

Specifies that only files which have been migrated from a user's local file system are to be exported.

### **Preview=***value*

Specifies whether you want to preview the results of the export operation, without actually exporting information. You can use this parameter to preview how many bytes of data are transferred, allowing you to determine how many volumes will be required to hold the data. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the node information is to be exported. If you specify this value, you must specify a value for the DEVCLASS parameter.

## EXPORT NODE

### Yes

Specifies that you want to preview the results of the export operation, without actually exporting node information. Information on the number and types of objects exported, together with the number of bytes transferred are reported to the server console and the activity log. If you specify this value, you do not need to specify device or volume parameters.

### **DEVclass=***devclassname*

Specifies the device class to which export data is to be written. This parameter is required unless you specified PREVIEW=YES. You cannot specify the DISK device class.

If the mount limit for the device class specified has already been reached (all drives are busy) when the export runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.

### **Scratch=***scratchvalue*

Specifies whether scratch volumes can be used for export. Scratch volumes are not supported for LAN3995 device classes. The default is SCRATCH=YES. Possible values are:

### Yes

Specifies that scratch volumes are to be used for export. If you include a list of volumes on the VOLUMENAMES parameter, then scratch volumes are used only if there is not enough space to store the exported files on the volumes specified. If the device type associated with the specified device class is FILE, a file named 00000001.EXP will be chosen for the first scratch volume. If a file already exists by this name in the target directory, you must first rename or erase it if you want to use SCRATCH=YES.

### No

Specifies that scratch volumes cannot be used for export. You must include a list of volumes on the VOLUMENAMES parameter to contain all of the export data. To determine how many volumes you may need, you can run the command specifying PREVIEW=YES.

### **VOLumentames=***volname*

Specifies one or more sequential volumes that are to be used to contain exported files. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. A volume name must be specified if SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, then scratch volumes are used.

Use the following naming conventions to specify volume names for each device type:

### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

## EXPORT NODE

### **4MM, 8MM, DLT, or QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.

### **OPTICAL or MO35**

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### **LAN3995**

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

## Examples

### **Task**

Export the node definitions and active backup data for all nodes defined in the STANDARD policy domain to 8MM tape (assuming a device class of MENU2).

The number and type of objects exported are reported to the system console and in the activity log.

**Command:** export node domains=standard devclass=menu2 scratch=Yes  
filedata=backupactive

### **Output:**

```
ANR0609I EXPORT NODE started as Process 12.  
ANS5104I Process number 12 started.
```

## EXPORT NODE

### Related Commands

Table 75 provides references to related commands.

*Table 75. Commands Related to EXPORT NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT ADMIN	Copies administrative information to external media.	Page 250
EXPORT POLICY	Copies policy information to external media.	Page 260
EXPORT SERVER	Copies all or part of the server to external media.	Page 264
IMPORT NODE	Restores client node information from external media.	Page 287
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## EXPORT POLICY

---

### EXPORT POLICY (Copy Policy Information to Sequential Media)

Use the EXPORT POLICY command to export policy information for one or more policy domains from the ADSM server to sequential media.

The EXPORT POLICY command exports policy information and client node data such as:

- Policy domain definitions
- Policy set definitions, including the active policy set
- Management class definitions, including the default management class
- Backup copy group and archive copy group definitions
- Schedule definitions and client node associations

The EXPORT POLICY command allows all policy information to be copied to another server. In order to back up ADSM data and then restore it to a new server, the sequential device must be supported on both server platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

This command is useful for permitting later retrieval of server information for copying the information a new server.

You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the export operation. You can also view this information from the server console.

This command generates a background process that can be canceled with the CANCEL PROCESS command. If an EXPORT POLICY background process is canceled, the sequential media holding the exported data are incomplete and should not be used for importing data. To display information on background processes, use the QUERY PROCESS command.

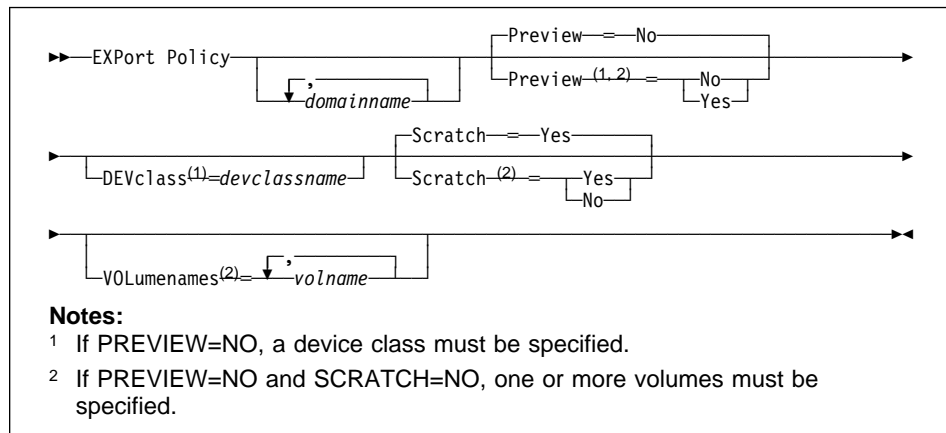
### Privilege Class

To issue this command, you must have system privilege.



## EXPORT POLICY

### Syntax



### Parameters

#### *domainname*

Specifies a list of policy domains for which policy information is to be exported. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use pattern-matching expressions to specify names. All matching policy domains are included in the list. If you do not specify a value for this parameter, information related to all policy domains is exported.

#### **Preview=value**

Specifies whether you want to preview the results of the export operation, without actually exporting information. You can use this parameter to preview how many bytes of data are transferred, allowing you to determine how many volumes will be required to hold the data. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the policy information is to be exported. If you specify this value, you must specify a value for the DEVCLASS parameter.

#### **Yes**

Specifies that you want to preview the results of the export operation, without actually exporting policy information. Information on the number and types of objects exported, together with the number of bytes transferred are reported to the server console and the activity log. If you specify this value, you do not need to specify device or volume parameters.

#### **DEVclass=devclassname**

Specifies the device class to which export data is to be written. This parameter is required unless you specified PREVIEW=YES. You cannot specify the DISK device class. If the mount limit for the device class specified has already been reached (all drives are busy) when the export runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.

## EXPORT POLICY

### **Scratch=***scratchvalue*

Specifies whether scratch volumes can be used for export. Scratch volumes are not supported for LAN3995 device classes. The default is SCRATCH=YES.

Possible values are:

#### **Yes**

Specifies that scratch volumes are to be used for export. If you include a list of volumes on the VOLUMENAMES parameter, then scratch volumes are used only if there is not enough space to store the exported files on the volumes specified. If the device type associated with the specified device class is FILE, a file named 00000001.EXP will be chosen for the first scratch volume. If a file already exists by this name in the target directory, you must first rename or erase it if you want to use SCRATCH=YES.

#### **No**

Specifies that scratch volumes cannot be used for export. You must include a list of volumes on the VOLUMENAMES parameter to contain all of the export data. To determine how many volumes you may need, you can run the command specifying PREVIEW=YES.

### **VOLumenames=***volname*

Specifies one or more tape volumes that are to be used to contain exported files. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. A volume name must be specified if SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, then scratch volumes are used.

For other device types, use the following naming conventions to specify volume names:

#### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

#### **4MM, 8MM, DLT, or QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

#### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.

#### **OPTICAL or MO35**

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

#### **LAN3995**

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

## EXPORT POLICY

### Examples

#### Task

Export the policy information related to all domains whose names begin with a D to a device class defined for 8MM tape (assuming a device class of CLASS4) and allowing for scratch volumes to be mounted.

The number and type of objects exported are reported to the system console and in the activity log.

**Command:** `export policy d* devclass=class4`

#### Output:

```
ANR0609I EXPORT POLICY started as process 22.  
ANS5104I Process number 22 started.
```

### Related Commands

Table 76 provides references to related commands.

*Table 76. Commands Related to EXPORT POLICY*

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT ADMIN	Copies administrative information to external media.	Page 250
EXPORT NODE	Copies client node information to external media.	Page 254
EXPORT SERVER	Copies all or part of the server to external media.	Page 264
IMPORT POLICY	Restores policy information from external media.	Page 293
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## EXPORT SERVER

---

### EXPORT SERVER (Copy Server Information to Sequential Media)

Use the EXPORT SERVER command to export all or part of the server control information and client file data (if specified) from the ADSM server to sequential media. You can also view this information from the server console.

The EXPORT SERVER command exports server control information such as:

- Policy domain definitions
- Policy set definitions
- Management class and copy group definitions
- Schedule definitions and client node associations
- Administrator definitions
- Client node definitions

Optionally, the EXPORT SERVER command exports server data information such as:

- File space definitions
- Data from files migrated from client nodes
- Archive or backup file copy data
- Access authorization information pertaining to the file spaces exported.

The EXPORT SERVER command allows all global server control information (such as policy data and administrative information) to be copied to another server. In order to back up ADSM data and then restore it to a new server, the sequential device must be supported on both platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

If a primary file cannot be exported because it is damaged or because it resides on a volume that is UNAVAILABLE or OFFLINE, ADSM attempts to obtain a copy of the file from a copy storage pool.

This command is useful for permitting later retrieval of server information for copying the information to a new server.

You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the export operation.

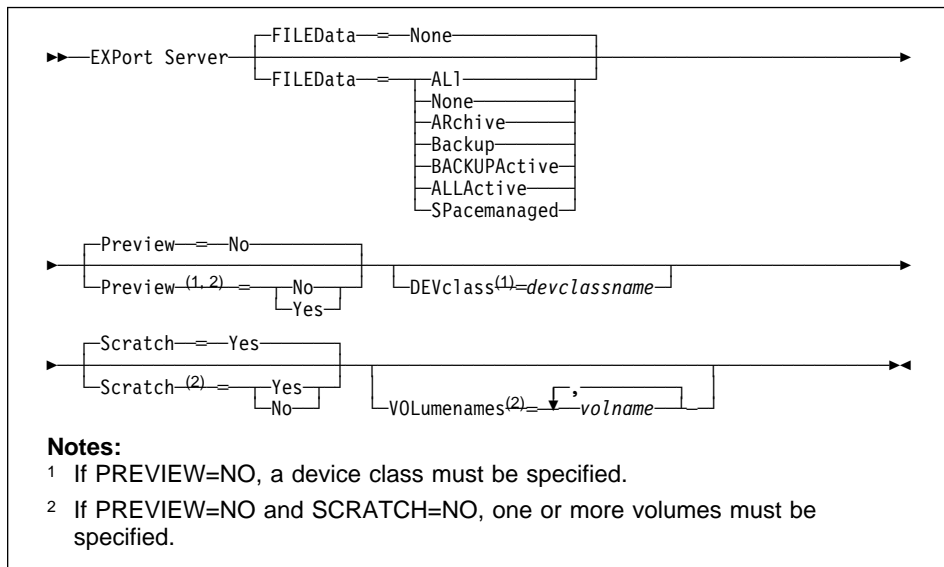
This command generates a background process that can be canceled with the CANCEL PROCESS command. If an EXPORT SERVER background process is canceled, the sequential media holding the exported data are incomplete and should not be used for importing data. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

## EXPORT SERVER

### Syntax



### Parameters

#### FILEData=*filedatatype*

Specifies the type of files that should be exported for all nodes defined to the server. This parameter is optional. The default value is NONE.

The device class used by the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, then two drives are needed to export server information. The mount limit value for the device class needs to be set to at least 2.

The following descriptions mention *active* and *inactive* backup file copies. An active backup file copy is the most recent backup copy for a file that still exists on the client workstation. All other backup file copies are called inactive copies. Possible values are:

#### ALI

Specifies that ADSM exports all backup versions of files, all archived files, and all files migrated from a user's local file system.

#### None

Specifies that no files are to be exported; only server definitions are to be exported.

#### ARchive

Specifies that only archived files are to be exported.

#### Backup

Specifies that only backup versions, whether active or inactive, are to be exported.

## EXPORT SERVER

### **BACKUPActive**

Specifies that only active backup versions are to be exported.

### **ALLActive**

Specifies that ADSM exports all active backup versions of files, all archived files, and all files migrated from a user's local file system.

### **SPacemanaged**

Specifies that only files which have been migrated from a user's local file system are to be exported.

### **Preview=*value***

Specifies whether you want to preview the results of the export operation, without actually exporting the server information. This command is used to determine how many bytes of data is transferred, allowing you to determine how many volumes will be required to hold the data. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the server information is to be exported. If you specify this value, you must specify a value for the DEVCLASS parameter.

#### **Yes**

Specifies that you want to preview the results of the export operation, without actually exporting server information. Information on the number and types of objects exported, together with the number of bytes transferred are reported to the server console and the activity log. If you specify this value, you do not need to specify device or volume parameters.

### **DEVclass=*devclassname***

Specifies the device class to which export data is to be written. This parameter is required unless you specified PREVIEW=YES. You cannot specify the DISK device class.

If the mount limit for the device class specified has already been reached (all drives are busy) when the export runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the export.

### **Scratch=*scratchvalue***

Specifies whether scratch volumes can be used for export. Scratch volumes are not supported for LAN3995 device classes. The default is SCRATCH=YES. Possible values are:

#### **Yes**

Specifies that scratch volumes are to be used for export. If you include a list of volumes on the VOLUMENAMES parameter, then scratch volumes are used only if there is not enough space to store the exported files on the volumes specified. If the device type associated with the specified device class is FILE, a file named 00000001.EXP will be chosen for the first scratch volume. If a file already exists by this name in the target directory, you must first rename or erase it if you want to use SCRATCH=YES.

## EXPORT SERVER

### No

Specifies that scratch volumes cannot be used for export. You must include a list of volumes on the VOLUMENAMES parameter to contain all of the export data. To determine how many volumes you may need, you can run the command specifying PREVIEW=YES.

### **VOL**umenames=*volname*

Specifies one or more tape volumes that are to be used to contain exported files. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. A volume name must be specified if SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, then scratch volumes are used.

Use the following naming conventions to specify volume names for each device type.

### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

### **4MM, 8MM, DLT, or QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.

### **OPTICAL or MO35**

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### **LAN3995**

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

## Examples

### Task

Before exporting the server, determine how many objects will be exported, and the total size in bytes required to export the entire server and all active backup and archived files.

The number and type of objects exported are reported to the server console and in the activity log.

**Command:** export server filedata=allactive preview=yes

## EXPORT SERVER

### Output:

```
ANR0609I EXPORT SERVER started as process 2.  
ANS5104I Process number 2 started.
```

### Related Commands

Table 77 provides references to related commands.

*Table 77. Commands Related to EXPORT SERVER*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT ADMIN	Copies administrative information to external media.	Page 250
EXPORT NODE	Copies client node information to external media.	Page 254
EXPORT POLICY	Copies policy information to external media.	Page 260
IMPORT SERVER	Restores all or part of the server from external media.	Page 297
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419



## EXTEND DB

---

### EXTEND DB (Increase the Assigned Capacity of the Database)

Use the EXTEND DB command to increase the amount of space that can be used by the database within all the database volumes previously allocated to ADSM.

**Attention:** The size of an allocated database, recovery log, or storage pool volume cannot be changed once it has been defined to the ADSM server. ADSM uses the initial size allocation of the volume at the time it is defined to the server to calculate data placement for later retrieval.

You can use this command after you define a database volume, via the DEFINE DBVOLUME command, to increase the assigned capacity of the database. To increase the capacity of the database, you must extend the database in 4MB increments. If you do not specify the extension in 4MB increments, ADSM rounds the number to the next 4MB partition. For example, if you extend the database by 1MB, ADSM extends the capacity of the database by 4MB.

The EXTEND DB command can be initiated while users are accessing the server. Use the QUERY DB command to determine how much you can increase the assigned capacity of the database.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶—EXTend DB—nnn————▶
```

### Parameters

*nnn*

Specifies the number of megabytes to be added to the assigned capacity of the database. This parameter is required. The value should be in increments of 4MB.

## EXTEND DB

### Examples

#### Task

Extend the assigned capacity of the database by 52MB.

**Command:** `extend db 52`

#### Output:

```
ANR2248I Database assigned capacity has been extended.
```

### Related Commands

Table 78 provides references to related commands.

*Table 78. Commands Related to EXTEND DB*

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
QUERY DB	Displays allocation information about the database.	Page 339
REDUCE DB	Attempts to free up a database volume by reducing usable storage within the volumes defined.	Page 457

## EXTEND LOG

---

### EXTEND LOG (Increase the Assigned Capacity of the Recovery Log)

Use the EXTEND LOG command to increase the amount of space that can be used by the recovery log within all the recovery log volumes previously allocated to ADSM.

**Attention:** The size of an allocated database, recovery log, or storage pool volume cannot be changed once it has been defined to the ADSM server. ADSM uses the initial size allocation of the volume at the time it is defined to the server to calculate data placement for later retrieval.

You can use this command after you define a recovery log volume, via the DEFINE LOGVOLUME command, to increase the assigned capacity of the recovery log. To increase the capacity of the recovery log, you must extend the recovery log in 4MB increments. If you do not specify the extension in 4MB increments, ADSM rounds the number to the next 4MB partition. For example, if you extend the recovery log by 1MB, ADSM extends the recovery log by 4MB.

The EXTEND LOG command can be initiated while users are accessing the server. Use the QUERY LOG command to determine how much you can increase the assigned capacity of the recovery log.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If an EXTEND LOG background process is canceled, the log may be partially extended. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶▶—EXTend LOG—nnn————▶▶
```

### Parameters

*nnn*

Specifies the number of megabytes to be added to the assigned capacity of the recovery log. This parameter is required. The value should be in increments of 4MB.

## EXTEND LOG

### Examples

#### Task

Extend the assigned capacity of the recovery log by 4MB.

**Command:** `extend log 4`

#### Output:

```
ANR2269I Recovery log extension process initiated
(process ID 5).
ANR0307I Recovery log extend in progress; 4 megabytes
of 4 formatted.
ANR2268I Recovery log assigned capacity has been extended.
```

### Related Commands

Table 79 provides references to related commands.

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
QUERY LOG	Displays allocation information about the recovery log.	Page 389
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
REDUCE LOG	Attempts to free up a log volume by reducing usable storage within the volumes defined.	Page 459

## GRANT AUTHORITY

### GRANT AUTHORITY (Add Administrator Authority)

Use the GRANT AUTHORITY command to grant an administrator one or more administrative privilege classes.

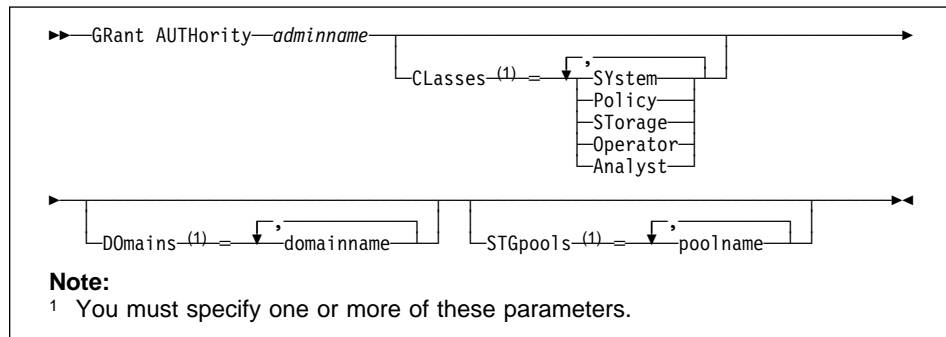
You can use this command to grant administrative privilege classes to an administrator that has not been granted any administrative privilege, or to add additional privilege classes to an administrator.

You cannot grant restricted privilege to an unrestricted policy or storage administrator. To do this, you must use the REVOKE AUTHORITY command to remove the administrator's unrestricted privilege, then use this command to grant restricted privilege to the administrator.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *adminname*

Specifies the name of the administrator being granted an administrative privilege class. This parameter is required.

#### **CLASSES**=*classes*

Specifies one or more privilege classes to be granted to an administrator. This parameter is optional when either the DOMAINS or STGPOLS parameters are specified; otherwise, this parameter is required. You can specify more than one privilege class by delimiting each with a comma. Possible classes are:

#### **S**ystem

Specifies that you want to grant system privilege to an administrator. Do not specify additional privilege classes or the DOMAINS or STGPOLS parameters when granting system privilege to an administrator. A system administrator has the highest level of authority in ADSM. A system administrator can issue any administrative command and has authority to

## GRANT AUTHORITY

manage all policy domains and all storage pools. Only a system administrator can grant authority to other administrators.

### **Policy**

Specifies that you want to grant policy privilege to an administrator. If the `DOMAINS` parameter is not specified, unrestricted policy privilege is granted. An unrestricted policy administrator can issue commands that affect all existing policy domains as well as any policy domains that are defined in the future. An unrestricted policy administrator cannot define, delete, or copy policy domains. A restricted policy administrator can be upgraded to an unrestricted policy administrator by using the `GRANT AUTHORITY` command with `CLASSES=POLICY` and no `DOMAINS` parameter.

### **STorage**

Specifies that you want to grant storage privilege to an administrator. If the `STGPOOLS` parameter is not specified, unrestricted storage privilege is granted. An unrestricted storage administrator can issue all commands that allocate and control storage resources for the server. An unrestricted storage administrator can issue commands that affect all existing storage pools as well as any storage pools that are defined in the future. An unrestricted storage administrator cannot define or delete storage pools. A restricted storage administrator can be upgraded to an unrestricted storage administrator by using the `GRANT AUTHORITY` command with `CLASSES=STORAGE` and no `STGPOOLS` parameter.

### **Operator**

Specifies that you want to grant operator privilege to an administrator. An administrator with operator privilege can issue commands that control the immediate operation of the server and the availability of storage media.

### **Analyst**

Specifies that you want to grant analyst privilege to an administrator. An administrator with analyst privilege can issue commands that reset the counters which track server statistics.

### **DOmains=domainname**

Specifies that you want to grant restricted policy privilege to an administrator. If the `DOMAINS` parameter is specified, then `CLASSES=POLICY` is optional.

Restricted policy privilege permits an administrator to issue a subset of the policy commands for the domains to which the administrator is authorized. You can use this parameter to grant additional policy domain authority to a restricted policy administrator. This parameter is optional. You can specify more than one policy domain by delimiting each policy domain name with a comma.

You can use a pattern-matching expression to specify a name. Authority for all matching policy domains is granted.

### **STGpools=poolname**

Specifies that you want to grant restricted storage privilege to an administrator. If the `STGPOOLS` parameter is specified, then `CLASSES=STORAGE` is optional.

## GRANT AUTHORITY

Restricted storage privilege permits you to issue a subset of the storage commands for the storage pools to which the administrator is authorized. You can use this parameter to grant additional storage pool authority to a restricted storage administrator. This parameter is optional. You can specify more than one storage pool by delimiting each storage pool name with a comma.

You can use a pattern-matching expression to specify a name. Authority for all matching storage pools is granted.

### Examples

#### Task 1

Grant system privilege to administrator LARRY.

**Command:** `grant authority larry classes=system`

**Output:**

```
ANR2076I System privilege granted to administrator LARRY.
```

#### Task 2

Specify additional policy domains that can be managed by the restricted policy administrator CLAUDIA.

**Command:** `grant authority claudia domains=employee_records,prog1`

**Output:**

```
ANR2078I Restricted policy privilege granted to
administrator CLAUDIA - policy domain EMPLOYEE_RECORDS.
ANR2078I Restricted policy privilege granted to
administrator CLAUDIA - policy domain PROG1.
```

#### Task 3

Provide administrator TOM with unrestricted storage privilege and restricted policy privilege for the domains whose names start with EMP.

**Command:** `grant authority tom classes=storage domains=emp*`

**Output:**

```
ANR2078I Restricted policy privilege granted to
administrator TOM - policy domain EMPLOYEE_RECORDS.
ANR2079I Unrestricted storage privilege granted
to administrator TOM.
```

## GRANT AUTHORITY

### Related Commands

Table 80 provides references to related commands.

*Table 80. Commands Related to GRANT AUTHORITY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
REVOKE AUTHORITY	Revokes one or more privilege classes or restricts access to policy domains and storage pools.	Page 498



## HALT (Shut Down the Server)

Use the HALT command to shut down the server. The HALT command forces an abrupt shutdown and results in the canceling of all administrative and client node sessions, even if they are not completed.

Any in-progress transactions interrupted by this command are rolled back when you restart the server. If the QUIESCE option is specified, transaction activity is halted when current transactions have completed, and then the server is shutdown.

The QUIESCE option is recommended *only* if you plan to perform a stand-alone database dump by using the DSMSERV DUMPDB command immediately following server shutdown. For additional information about the DSMSERV DUMPDB command and related commands, see Appendix A, "Miscellaneous Utilities" on page 633.

It is recommended to use the HALT command only after all administrative and client node sessions are completed or canceled. In order to shut down the server without severely impacting administrative and client node sessions, perform the following steps:

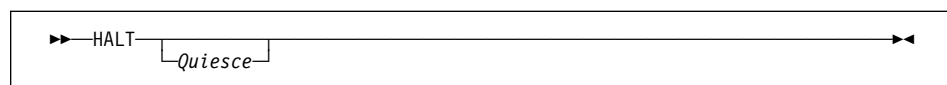
1. Use the DISABLE command to prevent new client node sessions from starting.
2. Use the QUERY SESSIONS command to identify any existing administrative and client node sessions.
3. Notify any existing administrative and client node sessions that you plan to shut down the server (you must do this outside of ADSM).
4. Use the CANCEL SESSIONS command to cancel any existing administrative or client node sessions.
5. Issue the HALT command to shut down the server and stop any administrative and client node sessions.

To start the server after issuing the HALT command, see the *ADSM Quick Start*.

## Privilege Class

To issue this command, you must have system privilege or operator privilege.

## Syntax



## Parameters

### Quiesce

Specifies that all database transaction activity should be halted when current transactions complete. This option should only be used when you are planning to perform a stand-alone database dump using the DSMSERV DUMPDB command to dump database contents to sequential media, such as tape.

## HALT

The default results with any in-progress transactions interrupted by this command are rolled back when you restart the server.

When using the QUIESCE option on the HALT command, the server attempts to stop all database update transactions against the database and record database pages to stable media so that a consistent image of the database can be dumped with the DSMSEV DUMPDB command. This attempt to stop update activity may be retried a number of times, and may even fail if there are many transactions in progress on the server. If this failure occurs, the HALT command stops processing and the server continues to run. If this occurs, you can cancel all client sessions, cancel all background server processes, and re-issue the HALT QUIESCE command. The HALT command without the QUIESCE option will always cause the server to halt successfully.

### Examples

#### Task 1

From the server console, shut down the server abruptly. All user activity stops immediately and no new activity is allowed to start.

**Command:** halt

**Output:**

```
ANR0991I ADSM server shut down complete.
```

#### Task 2

From an administrative client, shut down the server abruptly. All user activity stops immediately and no new activity is allowed to start.

**Command:** halt

**Output:**

```
ANR2234W This command will halt the server; if the command
is issued from a remote client, it may not be possible to
restart the server from the remote location.
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANS5103I Highest return code was 50.
```

## HALT

### Task 3

Before performing the stand-alone database dump by using the DSMSERV DUMPDB command, shut down the ADSM server with the QUIESCE option.

**Command:** halt quiesce

**Output:**

```
ANR2106I HALT: Quiescing database update activity.  
ANR2107I HALT: Database update activity is now quiesced.  
ANR0991I ADSM server shutdown complete.  
ANR5963I ADSM server termination complete.
```

The stand-alone dump can now be performed.

### Related Commands

Table 81 provides references to related commands.

*Table 81. Commands Related to HALT*

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
CANCEL SESSION	Cancels one or all active sessions with the server.	Page 78
DISABLE	Prevents new clients from accessing ADSM but permits existing sessions to continue.	Page 243
ENABLE	Resumes server activity following the DISABLE command.	Page 246
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY SESSION	Displays information about all active administrator and client sessions with ADSM.	Page 427

## HELP

---

### HELP (Get Help on Commands and Error Messages)

Use the HELP command to display information about administrative commands and error messages from the server and administrative command line client. For the best usability, issue the HELP command from an administrative command line client.

You can access the help facility by entering one of the following command strings:

- Help
- Help *help\_menu\_number*
- Help *commandname*
- Help *commandname subcommandname*
- Help *error\_msg\_number*

The HELP command (with no operands) displays a menu with a list of selections that provide information on using the command line interface and a list of the ADSM administrative commands.

The Help *help\_menu\_number* command displays the help text for a specific menu selection from the help menu.

The Help *commandname* command displays the help text specific to that command. If this command has subcommand names, you will also receive a list of the subcommand names. You can then select specific help related to the subcommand name.

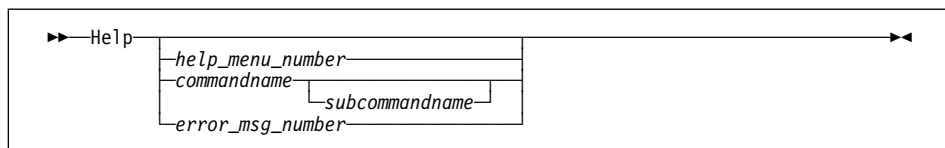
The Help *commandname subcommandname* command displays the help text specific to the command and its associated subcommands.

The Help *error\_msg\_number* command displays the help text specific to the error message. You can get help information on server messages (prefixed by ANR) and client messages (prefixed by ANS).

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

#### *help\_menu\_number*

Specifies the number of your selection from the help menu. This parameter is optional.

## HELP

### *commandname*

Specifies the name of the administrative command you want to display. This parameter is optional.

### *subcommandname*

Specifies the subcommand name associated with the name of administrative command that you want to display. This parameter is optional.

### *error\_msg\_number*

Specifies the number of the error message for which you want to display information. This parameter is optional. You can get help information on server messages (prefixed by ANR) and client messages (prefixed by ANS). You do not have to include the prefix and severity code when specifying an error message number.

## Examples

### Task 1

Display the help menu for the command-line interface.

**Command:** help

**Output:**

```
The following help sections are available:
1 - Using Syntax Diagrams
2 - Entering Administrative Commands
3 - Using the Server Console Session
4 - Using the Administrative Client
5 - Using the Administrative Client Options
6 - Using Continuation Characters
7 - Using Console Mode
8 - Using Mount Mode
9 - Using Batch Mode
10 - Using Interactive Mode
11 - Redirecting Command Output to a File
12 - Using Macros to Issue Commands
13 - Using Commands Based on Privilege Classes
14 - Commands Requiring System Privilege
15 - Commands Requiring Policy Privilege
16 - Commands Requiring Storage Privilege
17 - Commands Requiring Operator Privilege
18 - Commands Requiring Analyst Privilege
19 - Commands Any Administrator Can Issue
```

### Task 2

Display help information using the help menu in Example 1.

**Command:** help 19

## HELP

**Output:**

Information on commands that can be issued by any administrator is displayed.

**Task 3**

Display help information on the REMOVE commands.

**Command:** help remove

**Output:**

```
REMOVE
-----
Use the REMOVE commands to remove an object from ADSM.
The following is a list of REMOVE commands for ADSM:
    "REMOVE ADMIN (Delete an Administrator)"
    "REMOVE NODE (Delete a Client Node)"
```

**Task 4**

Display help information on the error message ANR7800I.

**Command:** Help 7800

**Output:**

```
ANR7800I  DSMSERV GENERATED AT time ON date.
EXPLANATION:  The ADSM server module was generated at the indicated date
time.
SYSTEM ACTION:  Server operation continues.
USER RESPONSE:  None.
```

### Related Commands

None.

## IMPORT ADMIN

### IMPORT ADMIN (Import Administrator Information)

Use the IMPORT ADMIN command to import administrator and authority definitions for one or more administrators from export media to the ADSM server.

The IMPORT ADMIN command imports administrator information such as:

- Administrator name, password, and contact information
- Administrator privilege classes the administrator has been granted

The IMPORT ADMIN command allows administrator information to be copied to another server. In order to copy ADSM data and then restore it to a new server, the sequential device must be supported on both platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

This command is useful for permitting retrieval of the server information so that the information can be copied to a new server.

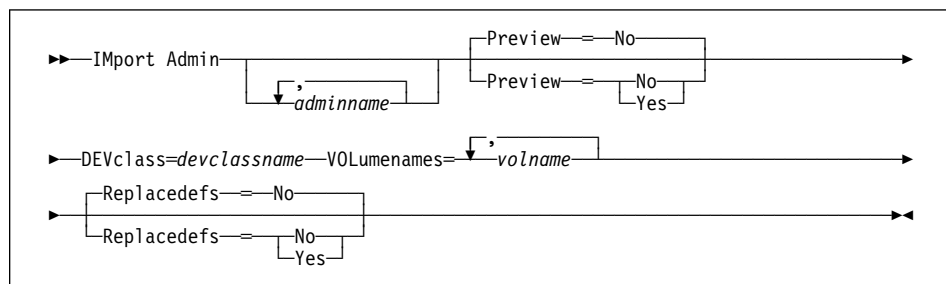
You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the import operation. You can also view this information from the server console.

This command generates a background process that can be canceled with the CANCEL PROCESS command. If an IMPORT ADMIN background process is canceled, some of the data is already imported. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *adminname*

Specifies the name of an administrator for which you want to import administrator information. This parameter is optional. You can specify a list of administrators for which administrator information is to be imported. The items in the list are

## IMPORT ADMIN

separated by commas, with no intervening spaces. You can use pattern-matching expressions to specify names. All matching administrators are included in the list. If you do not specify a value for this parameter, information related to all administrators found on the import media is imported.

### **Preview=*value***

Specifies whether you want to preview the results of the import operation, without actually importing administrator information. For the import commands, the PREVIEW=YES option will require that export volumes be mounted to generate the preview statistics. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the information is to be imported.

#### **Yes**

Specifies that you want to preview the results of the import operation, without actually importing the administrator information. Information on the number and types of objects imported, together with the number of bytes transferred are reported to the server console and the activity log.

### **DEVclass=*devclassname***

Specifies the device class from which import data is to be read. This parameter is required. You cannot specify the DISK device class.

If the mount limit for the device class specified has already been reached (all drives are busy) when the import runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the import.

### **VOLumentnames=*volname***

Specifies the name of a volume from which you want to import information. This parameter is required. You can specify a list of volumes from which server information is to be imported. The list of volumes needs to be in the same sequential order as they were exported in. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. The items in the list are separated by commas, with no intervening spaces.

Use the following naming conventions to specify volume names for each device type:

#### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

#### **4MM, 8MM, DLT, or QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

#### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.



## IMPORT ADMIN

### OPTICAL or MO35

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### LAN3995

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

### Replacedefs=*value*

Specifies whether to replace administrator definitions that already exist on the target server. The default value is NO. Possible values are:

#### No

Specifies that definitions are not to be replaced by imported objects.

#### Yes

Specifies that definitions are to be replaced by the imported objects.

**Note:** When specifying REPLACEDFS=YES, be sure you want to import data.

## Examples

### Task

From an OS/2 server, import the administrator information for all defined administrators from an 8MM tape (assuming a device class of MENU1).

The number and type of objects imported are reported to the system console and in the activity log.

**Command:** `import admin devclass=menu1`

### Output:

```
ANR0609I IMPORT ADMIN started as process 19.  
ANS5104I Process number 19 started.
```

## Related Commands

Table 82 provides references to related commands.

Table 82 (Page 1 of 2). Commands Related to IMPORT ADMIN

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT ADMIN	Copies administrative information to external media.	Page 250

## IMPORT ADMIN

*Table 82 (Page 2 of 2). Commands Related to IMPORT ADMIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
IMPORT NODE	Restores client node information from external media.	Page 287
IMPORT POLICY	Restores policy information from external media.	Page 293
IMPORT SERVER	Restores all or part of the server from external media.	Page 297
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## IMPORT NODE

---

### IMPORT NODE (Import Client Node Information)

Use the IMPORT NODE command to import client node definitions from the ADSM server to sequential media. Client node definitions include information such as:

- User ID, password, and contact information
- Name of the policy domain to which the client belongs
- File compression status
- Whether the user has the authority to delete backed up or archive files from data storage
- Whether the client node ID was locked at the time of export

Optionally, the IMPORT NODE command imports node data information such as:

- File space definitions
- Data from files migrated from HSM client nodes
- Archive or backup file copy data
- Access authorization information pertaining to the file spaces imported.

The IMPORT NODE command allows node information to be copied to another server. In order to copy ADSM data and then restore it to a new server, the sequential device must be supported on both platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

This command is useful for permitting retrieval of server information for copying the information to a new server.

You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the import operation. You can also view this information from the server console.

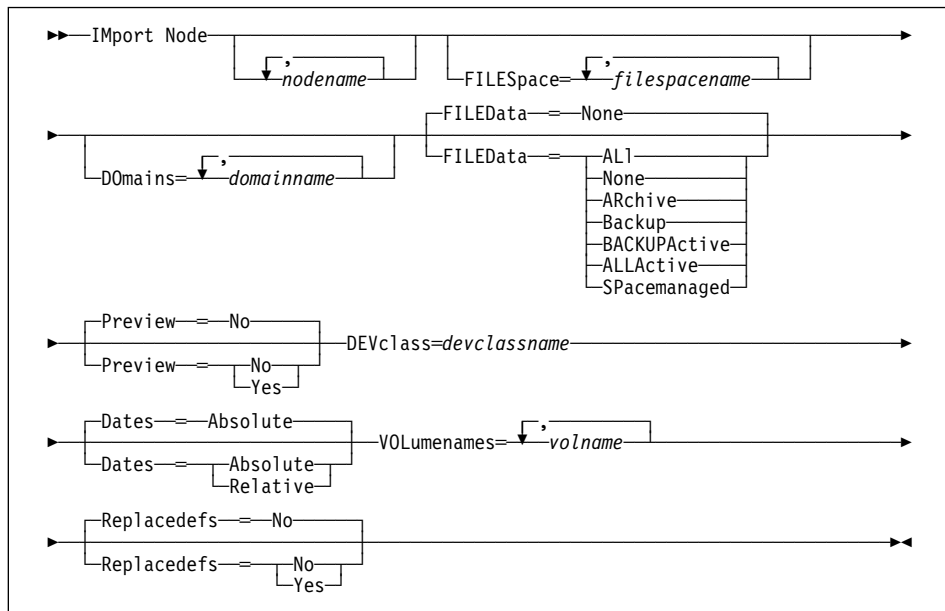
This command generates a background process that can be canceled with the CANCEL PROCESS command. If an IMPORT NODE background process is canceled, some of the data is already imported. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

## IMPORT NODE

### Syntax



### Parameters

#### *nodename*

Specifies the name of a client node for which you want to import client node information. This parameter is optional. You can specify a list of node names for which node information is to be imported. The items in the list are separated by commas, with no intervening spaces. You can use pattern-matching expressions to specify names. All matching nodes are included in the list. If you do not specify a value for this parameter, information related to all nodes is imported.

#### **FILESpace=filespace**

Specifies the name of a file space for which you want import file space information. This parameter is optional.

You can specify a list of file space names for which data information is to be imported. The items in the list are separated by commas, with no intervening spaces. You can use a pattern-matching expression to specify a name. All matching names are included in the list. If you do not specify a value for this parameter, information related to all files space names is imported.

**Note:** Existing file spaces are not replaced. New file spaces are created when identical names are encountered.

#### **DOMAINS=domainname**

Specifies the names of the policy domains to which nodes must be assigned in order for node information to be imported. This parameter is optional. You can specify a list of policy domains from which node information is to be imported. The items in the list are separated by commas, with no intervening spaces. You

## IMPORT NODE

can use a pattern-matching expression to specify a name. All matching domains are included in the list. If you do not specify a value for this parameter, node information related to all domains is imported.

### **FILEData=type**

Specifies the type of files that should be imported for all nodes specified and found on the export media. This parameter is optional. The default value is NONE.

The device class used by the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, then two drives are needed to import the node information. The mount limit value for the device class needs to be set to at least 2.

The following descriptions mention *active* and *inactive* backup file copies. An active backup file copy is the most recent backup copy for a file that still exists on the client workstation. All other backup file copies are called inactive copies. Possible values are:

#### **ALI**

Specifies that ADSM imports all backup versions of files, all archived files, and all files migrated from a user's local file system.

#### **None**

Specifies that no files are to be imported; only node definitions are to be imported.

#### **ARchive**

Specifies that only archived files are to be imported.

#### **Backup**

Specifies that only backup versions, whether active or inactive, are to be imported.

#### **BACKUPActive**

Specifies that only active backup versions are to be imported.

#### **ALLActive**

Specifies that ADSM imports all active backup versions of files, all archived files, and all files migrated from a user's local file system.

#### **SPacemanaged**

Specifies that only files that have been migrated from a user's local file system are to be imported.

## IMPORT NODE

### **Preview=***value*

Specifies whether you want to preview the results of the import operation, without actually importing information. For the import commands, the PREVIEW=YES option will require that all export volumes be mounted to generate the preview statistics. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the node information is to be imported.

#### **Yes**

Specifies that you want to preview the results of the import operation, without actually importing files. Information on the number and types of objects imported, together with the number of bytes transferred are reported to the server console and the activity log.

### **DEVclass=***devclassname*

Specifies the device class from which import data is to be read. You cannot specify the DISK device class. This parameter is required.

If the mount limit for the device class specified has already been reached (all drives are busy) when the import runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the import operation.

### **Dates=***value*

Specifies whether the recorded backup or archive dates for client node file copies are set to the values specified when the files were exported (Absolute), or are adjusted relative to the date of import (Relative). The default value is ABSOLUTE.

If the export media has been idle for some time after export (sitting on a shelf for six months, for example), the original backup or archive dates may be old enough to trigger immediate roll-off of file copies when the data is imported into a server. The Relative specification for this value will adjust for time elapsed since export so that the file copies are not immediately rolled off.

For example, assume that an export tape contains an archive file copy that was archived 5 days prior to the export operation. If the export media is saved for 6 months and then imported, the archive file will appear to have been inserted six months and 5 days ago by default (Dates=Absolute) and may be subject to immediate roll-off depending upon the retention value that is specified in the file's management class. Specifying Dates=Relative will result in resetting the archive date for the file to 5 days ago during import, even though the actual file was archived six months and 5 days ago. The DATES=RELATIVE parameter thus adjusts file backup and archive dates for the time that has elapsed since the export operation occurred.

Possible values are:

#### **ABSOLUTE**

Specifies the date for file copies are set to the values specified when the files were exported.

## IMPORT NODE

### RELATIVE

Specifies the date for file copies are adjusted to the date of import.

### **VOL**umenames=*volname*

Specifies the name of a volume from which you want to import volume information. This parameter is required. You can specify a list of volumes from which server information is to be imported. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. The items in the list are separated by commas, with no intervening spaces.

Use the following naming conventions to specify volume names for each device type:

### REMOVABLEFILE

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

### 4MM, 8MM, DLT, or QIC

Specify 1 to 6 alphanumeric characters, for example DSMT01.

### FILE

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.

### OPTICAL or MO35

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### LAN3995

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

### Replacedefs=*value*

Specifies whether node definitions that exist on the server should be replaced by imported objects having the same name. Existing file spaces are not replaced. New file spaces are created when identical file space names are encountered. The default value is NO. Possible values are:

#### No

Specifies that objects are not to be replaced by imported objects.

#### Yes

Specifies that objects are to be replaced by the imported objects.

## Examples

### Task

From a server, import the node definitions and active backup data for all nodes defined in the STANDARD policy domain from a file (assuming a device class of DEPT).

The number and type of objects imported are reported to the server console and in the activity log.

## IMPORT NODE

**Command:** import node domains=standard devclass=dept filedata=backupactive

**Output:**

```
ANR0609I IMPORT NODE started as process 32.  
ANS5104I Process number 32 started.
```

### Related Commands

Table 83 provides references to related commands.

*Table 83. Commands Related to IMPORT NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT NODE	Copies client node information to external media.	Page 254
IMPORT ADMIN	Restores administrative information from external media.	Page 283
IMPORT POLICY	Restores policy information from external media.	Page 293
IMPORT SERVER	Restores all or part of the server from external media.	Page 297
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419



## IMPORT POLICY

### IMPORT POLICY (Import Policy Information)

Use the IMPORT POLICY command to import policy information for one or more policy domains from export media to the ADSM server.

The IMPORT POLICY command imports policy information such as:

- Policy domain definitions
- Policy set definitions, including the active policy set
- Management class definitions, including the default management class
- Backup copy group and archive copy group definitions
- Schedule definitions and client node associations

The IMPORT POLICY command allows policy information to be copied to another server. In order to back up ADSM data and then restore it to a new server, the sequential device must be supported on both platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

This command is useful for permitting retrieval of the server information for copying the information to a new server.

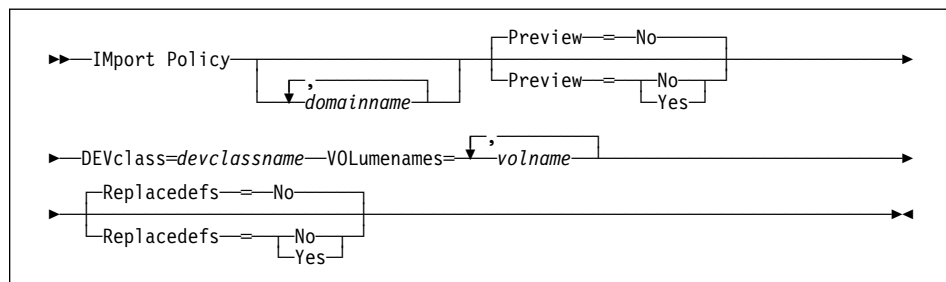
You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the import operation. You can also view this information from the server console.

This command generates a background process that can be canceled with the CANCEL PROCESS command. If an IMPORT POLICY background process is canceled, some of the data is already imported. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



## IMPORT POLICY

### Parameters

#### *domainname*

Specifies a list of policy domains for which policy information is to be imported. If you specify more than one policy domain, separate the names with commas, with no intervening spaces. You can use pattern-matching expressions to specify names. The default (\*) is to import the information related to all policy domains defined on the export media.

#### **Preview=***value*

Specifies whether you want to preview the results of the import operation, without actually importing information. For the import commands, the PREVIEW=YES option will require that all export volumes be mounted to generate the preview statistics. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the information is to be imported.

#### **Yes**

Specifies that you want to preview the results of the import operation, without actually importing files. Information on the number and types of objects imported, together with the number of bytes transferred are reported to the server console and the activity log.

#### **DEVclass=***devclassname*

Specifies the device class from which import data is to be read. You cannot specify the DISK device class. This parameter is required.

If the mount limit for the device class specified has already been reached (all drives are busy) when the import runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the import.

#### **VOL***names=volname*

Specifies the name of a volume from which you want to import volume information. This parameter is required. You can specify a list of volumes from which server information is to be imported. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. The items in the list are separated by commas, with no intervening spaces.

Use the following naming conventions to specify volume names for each device type:

#### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

#### **4MM, 8MM, DLT, or QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

#### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example \exadata\mt1.

## IMPORT POLICY

### OPTICAL or MO35

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### LAN3995

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, GSLMA1/GSAMA2).

### Replacedefs=*value*

Specifies whether objects (such as domains, policy sets) that exist on the server should be replaced by imported objects having the same name. The default value is NO. Possible values are:

#### Yes

Specifies that objects are to be replaced by the imported objects.

#### No

Specifies that objects are not to be replaced by imported objects.

## Examples

### Task

Import the policy information related to all domains whose names begin with a D from 3 volumes (EXP001, EXP002, EXP003) on an 8mm tape (assuming a device class of 8MM).

The number and type of objects imported are reported to the system console and in the activity log.

**Command:** `import policy d* devclass=8mm volumenames=exp001,exp002,exp003`

### Output:

```
ANR0609I IMPORT POLICY started as process 21.  
ANS5104I Process number 21 started.
```

## Related Commands

Table 84 provides references to related commands.

Table 84 (Page 1 of 2). Commands Related to IMPORT POLICY

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT POLICY	Copies policy information to external media.	Page 260

## IMPORT POLICY

*Table 84 (Page 2 of 2). Commands Related to IMPORT POLICY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
IMPORT ADMIN	Restores administrative information from external media.	Page 283
IMPORT NODE	Restores client node information from external media.	Page 287
IMPORT SERVER	Restores all or part of the server from external media.	Page 297
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## IMPORT SERVER

---

### IMPORT SERVER (Import the Server)

Use the IMPORT SERVER command to copy all or part of the server control information and client file data (if specified) from export media to the ADSM server. You can also view this information from the server console.

The IMPORT SERVER command imports server control information and client node data such as:

- Policy domain definitions
- Policy set definitions
- Management class and copy group definitions
- Schedule definitions and node associations
- Administrator definitions
- Client node definitions

Optionally, the IMPORT SERVER command imports server data information such as:

- File space definitions
- Data from files migrated from HSM client nodes
- Archive or backup file copy data
- Access authorization information pertaining to the file spaces exported.

The IMPORT SERVER command allows all global server control information (such as policy data and administrator information) to be copied to another server. In order to back up ADSM data and then restore it to a new server, the sequential device must be supported on both platforms.

ADSM client data can be moved between ADSM servers via export and import processing, if the same removable media type is supported on both platforms.

This command is useful for permitting retrieval of the server information for copying the information to a new server.

You can use the QUERY ACTLOG command to view the actual status information indicating the size and the success or failure of the import operation.

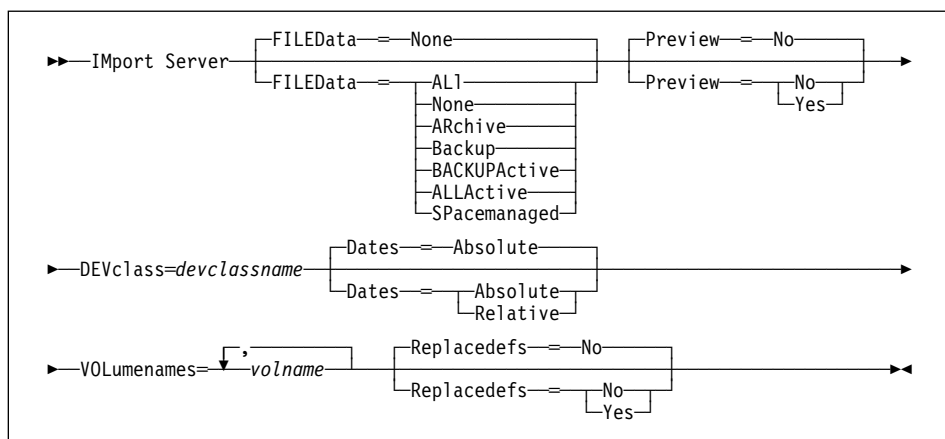
This command generates a background process that can be canceled with the CANCEL PROCESS command. If an IMPORT SERVER background process is canceled, some of the data is already imported. To display information on background processes, use the QUERY PROCESS command.

## IMPORT SERVER

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### **FILEData=type**

Specifies the type of files that should be imported for all nodes defined to the server. This parameter is optional. The default value is `NONE`.

The device class used by the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, then two drives are needed to import server information. The mount limit value for the device class needs to be set to at least 2.

The following descriptions mention *active* and *inactive* backup file copies. An active backup file copy is the most recent backup copy for a file that still exists on the client workstation. All other file copies are called inactive copies. Possible values are:

#### **ALI**

Specifies that ADSM imports all backup versions of files, all archived files, and all files migrated from a user's local file system.

#### **None**

Specifies that no files are to be imported; only server definitions are to be imported.

#### **ARchive**

Specifies that only archived files are to be imported.

#### **Backup**

Specifies that only backup versions, whether active or inactive, are to be imported.

## IMPORT SERVER

### **BACKUPActive**

Specifies that only active backup versions are to be imported.

### **ALLActive**

Specifies that ADSM imports all active backup versions of files, all archived files, and all files migrated from a user's local file system.

### **SPacemanaged**

Specifies that only files which have been migrated from a user's local file system are to be imported.

### **Preview=*value***

Specifies whether you want to preview the results of the import operation, without actually importing server information. For the import commands, the PREVIEW=YES option will require that all export volumes be mounted to generate the preview statistics. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the server information is to be imported.

#### **Yes**

Specifies that you want to preview the results of the import operation, without actually importing files. Information on the number and types of objects imported, together with the number of bytes transferred are reported to the server console and the activity log.

### **DEVclass=*devclassname***

Specifies the device class from which import data is to be read. You cannot specify the DISK device class. This parameter is required.

If the mount limit for the device class specified has already been reached (all drives are busy) when the import runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the import.

### **Dates=*value***

Specifies whether the recorded backup or archive dates for client node file copies are set to the values specified when the files were exported (Absolute), or are adjusted relative to the date of import (Relative). The default value is ABSOLUTE.

If the export media has been idle for some time after export (sitting on a shelf for six months, for example), the original backup or archive dates may be old enough to trigger immediate roll-off of file copies when the data is imported into a server. The Relative specification for this value will adjust for time elapsed since export so that the file copies are not immediately rolled off.

For example, assume that an export tape contains an archive file copy that was archived 5 days prior to the export operation. If the export media is saved for 6 months and then imported, the archive file will appear to have been inserted six months and 5 days ago by default (Dates=Absolute) and may be subject to immediate roll-off depending upon the retention value that is specified in the file's management class. Specifying DATES=RELATIVE results in resetting the

## IMPORT SERVER

archive date for the file to 5 days ago during import, even though the actual file was archived six months and 5 days ago. `DATES=RELATIVE` parameter thus adjusts file backup and archive dates for the time that has elapsed since the export operation occurred.

Possible values are:

### **ABSOLUTE**

Specifies the date for file copies are set to the values specified when the files were exported.

### **RELATIVE**

Specifies the date for file copies are adjusted to the date of import.

### **VOLumentnames=volname**

Specifies the name of a volume from which you want to import volume information. This parameter is required. You can specify a list of volumes from which server information is to be imported. Any volume name used by ADSM cannot contain any embedded blanks or equal signs. The items in the list are separated by commas, with no intervening spaces.

Use the following naming conventions to specify volume names for each device type:

### **REMOVABLEFILE**

Specify 1 to 8 alphanumeric characters, with no embedded blanks or periods.

### **4MM, 8MM, DLT, or QIC**

Specify 1 to 6 alphanumeric characters, for example DSMT01.

### **FILE**

Specify any supported file name, directory, or fully qualified directory, for example `\exadata\mt1`.

### **OPTICAL or MO35**

Specify 1 to 32 alphanumeric characters, for example OPTDST1.

### **LAN3995**

Specify two fully qualified alphanumeric names, each having up to 32 characters, separated by a forward slash (for example, `GSLMA1/GSAMA2`).

### **Replacedefs=value**

Specifies whether objects (such as domains, policy sets) that exist on the server should be replaced by imported objects having the same name. Existing file spaces are not replaced. New file spaces are created when identical names are encountered. The default value is NO. Possible values are:

### **No**

Specifies that objects are not to be replaced by imported objects.



## IMPORT SERVER

**Yes**

Specifies that objects are to be replaced by the imported objects.

### Examples

**Task**

Before importing the server control information, determine how many objects are imported and the total number of bytes copied to import the server control information and all archived files from three volumes (exp001,exp002,exp003) on 8MM tape.

The number and type of objects imported is reported to the server console and in the activity log.

**Command:** `import server filedata=archive devclass=8mm  
volumenames=exp001,exp002,exp003 preview=yes`

**Output:**

```
ANR0609I IMPORT SERVER started as process 2.  
ANS5104I Process number 2 started.
```

## IMPORT SERVER

### Related Commands

Table 85 provides references to related commands.

*Table 85. Commands Related to IMPORT SERVER*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
EXPORT SERVER	Copies all or part of the server to external media.	Page 264
IMPORT ADMIN	Restores administrative information from external media.	Page 283
IMPORT NODE	Restores client node information from external media.	Page 287
IMPORT POLICY	Restores policy information from external media.	Page 293
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## LOCK ADMIN

---

### LOCK ADMIN (Lock Out an Administrator)

Use the LOCK ADMIN command to prevent an administrator from accessing the server. By using the LOCK ADMIN and UNLOCK ADMIN commands, you can temporarily restrict an administrator from accessing the server. The administrator is locked out until a system administrator uses the UNLOCK ADMIN command to reestablish access for the administrator.

You cannot issue the LOCK ADMIN command against the SERVER\_CONSOLE administrative ID.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
▶▶—LOCK Admin—adminname————▶▶
```

### Parameters

*adminname*

Specifies the name of the administrator to be locked out. This parameter is required.

### Examples

#### Task

Lock out the administrator CLAUDIA.

**Command:** lock admin claudia

#### Output:

```
ANR2072I Administrator CLAUDIA locked.
```

### Related Commands

Table 86 provides references to related commands.

Table 86. Commands Related to LOCK ADMIN

Command	Description	Reference
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
UNLOCK ADMIN	Enables a locked administrator to access ADSM.	Page 535

## LOCK NODE

---

### LOCK NODE (Lock Out a Client Node)

Use the LOCK NODE command to prevent a client node from accessing the server.

By using the LOCK NODE command, you can restrict a client node from accessing the server. The client node is locked out until a system administrator uses the UNLOCK NODE command to reestablish access for the client node.

A locked client cannot perform backup and archive operations, regardless of whether the operations have been scheduled to occur.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node belongs.

### Syntax

```
▶▶—LOCK Node—nodename—▶▶
```

### Parameters

*nodename*

Specifies the name of the client node to be locked out. This parameter is required.

### Examples

#### Task

Lock the client node SMITH.

**Command:** lock node smith

#### Output:

```
ANR2064I Node SMITH locked.
```

## LOCK NODE

### Related Commands

Table 87 provides references to related commands.

*Table 87. Commands Related to LOCK NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403
UNLOCK NODE	Enables a locked user in a specific policy domain to access the server.	Page 536

## MACRO

---

### MACRO (Invoke a Macro)

Use the MACRO command to invoke a file that contains one or more ADSM administrative commands to be performed.

**Note:** You cannot schedule MACRO or QUERY ACTLOG commands.

You cannot run macros from the server console. You can include MACRO commands within a macro to invoke other macros up to ten levels deep. A macro invoked from the batch or interactive mode of the administrative client is called a *high-level* macro. Any macros invoked from within the high-level macro are called *nested* macros.

Writing commands in a macro can be especially helpful when you want to issue commands that contain several parameters. You might also want to write macros to process related commands in a specific order. After you create a macro, you can update the information it contains and use it again, or you can copy the macro, make changes to the copy, then run the copy.

To document its purpose and function, add comments to your macro file by entering a slash and an asterisk (/\*), which identifies the beginning of the comment. Then enter the comment and an asterisk and a slash (\*/) to indicate the end of the comment. You can enter a comment on a line by itself, or enter it on a line that contains a command or part of a command. You cannot nest comments or span comments on more than one line. For more information, see “Using Macros to Issue Commands” on page 21.

If you have a series of commands that process successfully via the command-line, but are unsuccessful when issued within a macro, there are probably dependencies between commands. It is possible that a command issued within a macro cannot be processed successfully until a previous command, that is issued within the same macro, is committed. Either of the following actions allow successful processing of these commands within a macro:

- Insert a COMMIT command before the command dependent on a previous command. For example, if COMMAND C is dependent upon COMMAND B you would insert a COMMIT command before COMMAND C. An example of this macro is:

```
COMMAND A  
COMMAND B  
COMMIT  
COMMAND C
```

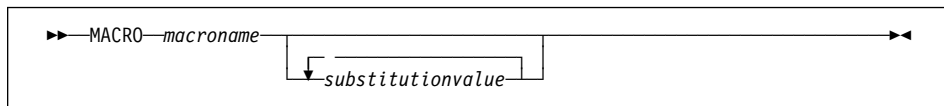
- Start the administrative client session using the ITEMCOMMIT option. This causes each command within a macro to be committed before the next command is processed.

### Privilege Class

Any administrator can issue this command.

## MACRO

### Syntax



### Parameters

#### *macroname*

Specifies the name of the macro. This parameter is required.

#### *substitutionvalue*

Specifies the value for a substitution variable in a macro. This parameter is optional. You can enter values for substitution variables to run a macro with substitution variables. When you use substitution variables, you can reuse a macro whenever you need to perform the same task for different objects or with different parameter values. Each substitution value must be delimited with a space. If you want to specify a value that contains blanks, you must enclose the value in quotation marks.

### Examples

#### Task 1

Use the macro file REGNG to register and grant authority to a new administrator.

#### Macro Contents:

```
/* Register and grant authority to new administrator */
REGister Admin jones passwd      -
CONtactinfo="x1235"
GRant AUTHority jones           -
CLasses=Policy
```

**Command:** macro regng

#### Output:

```
ANS5101I Server command: 'REGister Admin jones passwd
contact=x1235'
ANR2068I Administrator JONES registered.

ANS5101I Server command: 'GRant AUTHority jones
CLasses=policy'
ANR2077I Unrestricted policy privilege granted to
administrator JONES.
```

#### Task 2

Use the macro file AUTHRG, containing substitution variables, to register and grant authority to a new administrator.

## MACRO

When you run the macro, you then enter the values you want to pass to the server to process the command.

### Macro Contents:

```
/* Registers and authorizes a new administrator */
REGister Admin %1 %2 - /* Enter userid and password */
CONtact=%3 /* Enter contact info (in quotes if nec.) */
GRant AUTHority %1 - /* Server uses variable already */
- /* defined by you */
CLasses=%4 /* Enter the privilege class */
```

**Command:** macro authrg jones passwd x1235 Policy

### Output:

```
ANS5101I Server command: 'REGISTER ADMIN jones passwd
CONtactinfo=x1235'
ANR2068I Administrator JONES registered.

ANS5101I Server command: 'GRANT AUTHority jones
CLasses=Policy'
ANR2077I Unrestricted policy privilege granted to
administrator JONES.
```

## Related Commands

Table 88 provides references to related commands.

Table 88. Commands Related to MACRO

Command	Description	Reference
COMMIT	Makes changes to the database permanent.	Page 87
ROLLBACK	Discards any uncommitted changes to the database since the last COMMIT was executed.	Page 503



## MOVE DATA

---

### MOVE DATA (Move Files on a Storage Pool Volume)

Use the MOVE DATA command to move data from either a primary or copy storage pool volume to other available storage pool volumes.

#### Restrictions:

1. You cannot move data from a primary storage pool volume to a copy storage pool.
2. You cannot move data on a copy storage pool volume to a different storage pool.

For additional information about storage pools, refer to *ADSM Administrator's Guide*.

Multiple sequential access volume mounts are required to move files that span volumes.

When moving files off of a random access volume, any cached copies of files on the volume (assuming caching is enabled) are erased. The MOVE DATA command is the only way to manually remove cached random access volume copies. (ADSM automatically removes cached files when the space they occupy is needed for primary files.)

If you are attempting to move files to volumes within the same storage pool, sufficient space must be available on the volumes or the operation fails.

You can use this command to move files from an offsite volume in a copy storage pool. Because the offsite volume cannot be mounted, the server obtains the active files on the offsite volume from either a primary storage pool or another copy storage pool. These files are then written to the destination volumes in the original copy storage pool.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If a MOVE DATA background process is canceled, some files may have already moved prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

Do not use the MOVE DATA command if a restore process (RESTORE STGPOOL or RESTORE VOLUME) is running. The MOVE DATA command could cause the restore to be incomplete.

### Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool to which the volume belongs and also for the new storage pool, if one is specified.

### Syntax

```
▶▶ MOVE Data -volname [STGpool] -stgpoolname ▶▶
```

## MOVE DATA

### Parameters

*volname*

Specifies the name of the storage pool volume where the file to be moved currently resides. This parameter is required.

**STGpool=stgpoolname**

Specifies the primary storage pool name to which you want to move files from a primary storage pool volume. This parameter is optional and does not apply to copy storage pool volumes. If you do not specify a value for this parameter, files are moved to other volumes within the same storage pool.

### Examples

#### Task

Move files from storage pool volume STGVOL.1 to any available volumes assigned to 8MMPOOL.

**Command:** `move data stgvol.1 stgpool=8mmpool`

#### Output:

```
ANR1140I Move Data process started for volume  
d:\home\build\stgvol.1 (process ID 205).
```

### Related Commands

Table 89 provides references to related commands.

*Table 89. Commands Related to MOVE DATA*

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## QUERY ACTLOG

### QUERY ACTLOG (Search Activity Log for Messages)

Use the QUERY ACTLOG command to search the server activity log for messages, selected by a range of time or date, message number, or string expression. If you do not specify any parameters for this command, all messages that were generated in the last hour are displayed.

**Note:** You cannot schedule the QUERY ACTLOG command using the DEFINE SCHEDULE command (central scheduling).

The activity log contains all messages that are sent to the server console under normal operation. These messages include session start and termination, background process messages, sequential access media mount information, and inventory expiration. Responses to commands entered at the server console are not recorded in the activity log unless the command affects or starts a background process or client session. In addition, all error messages are retained in the activity log.

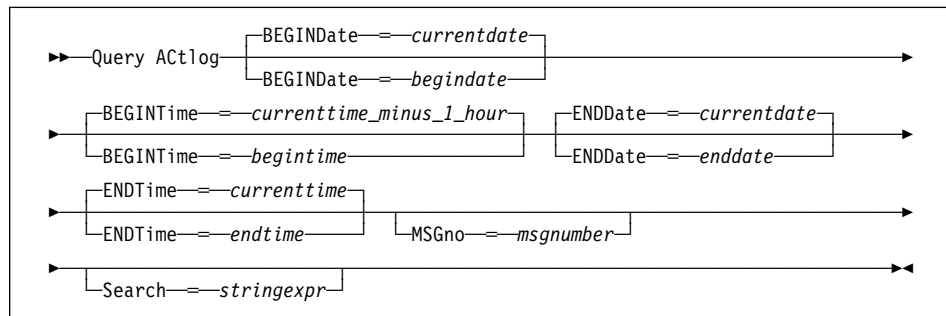
To minimize the processing time when querying the activity log, you should:

- Minimize the length of the time range for querying messages
- Restrict the message number or text string to those for which information is required

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

#### **BEGINDate=***date*

Specifies the beginning date of the range for the messages to be displayed, in MM/DD/YYYY format. This parameter is optional. All messages, meeting the time range criteria, that occurred after this date are displayed. The default value is the current date. Possible values are:

#### *MM/DD/YYYY*

Specifies that you want to display information beginning with records created on a specific date, using MM/DD/YYYY format (for example, 01/31/1995).

## QUERY ACTLOG

### **TODAY**

Specifies that you want to display information beginning with records created on the current date.

### **TODAY**-*numdays* or *-numdays*

Specifies that you want to display information beginning with records created on the current date minus a specific number of days. The maximum value for the number of days is 9999.

For example, to display information beginning with records created a week ago, you can specify TODAY-7 or simply -7.

### **BEGINTime**=*currenttime\_minus\_1\_hour*

Specifies the beginning time of the range for messages to be displayed, in HH:MM format. All messages, meeting the time range criteria, that occurred after this time are displayed. This parameter is optional. If you do not specify a value for this parameter, all messages that have occurred in the last hour are displayed.

Possible values are:

#### *HH:MM:SS*

Specifies that you want to display information beginning with records created at a specific time on the specified begin date using HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that you want to display information beginning with records created on or before the current time on the specified begin date.

### **NOW**+*numhours:numminutes* or *+numhours:numminutes*

Specifies that you want to display information beginning with records created on or before the current time plus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue the QUERY ACTLOG command at 9:00 with BEGINTime=NOW+3 or BEGINTime=+3, ADSM displays records with a time of 12:00 or later on the specified begin date.

### **NOW**-*numhours:numminutes* or *-numhours:numminutes*

Specifies that you want to display information beginning with records created on or before the current time minus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue the QUERY ACTLOG command at 9:00 with BEGINTime=NOW-3:30 or BEGINTime=-3:30, ADSM displays records with a time of 5:30 or later on the specified begin date.

### **ENDDate**=*date*

Specifies the ending date of the range for messages to be displayed, in MM/DD/YYYY format. This parameter is optional. All messages, meeting the time range criteria, that have occurred before this date are displayed. If you do not specify a value for this parameter, the current date is used. All the dates and times used are server date and time. Possible values are:

## QUERY ACTLOG

### *MM/DD/YYYY*

Specifies that you want to display information ending with records created on a specific date using MM/DD/YYYY format (for example, 02/28/95).

### **TODAY**

Specifies that you want to display information ending with records created on the current date.

### **TODAY**-*numdays* or *-numdays*

Specifies that you want to display information ending with records created on or before the current date minus a specific number of days. The maximum value for the number of days is 9999.

For example, to display all volume history information created up to yesterday, you can specify TODAY-1 or simply -1.

### **ENDTime**=*currenttime*

Specifies the ending time of the range for messages to be displayed, in HH:MM format. All messages, meeting the time range criteria, that have occurred before this time are displayed. This parameter is optional. If you do not specify a value for this parameter, all messages are displayed up to the time that you issued this command.

Possible values are:

### *HH:MM:SS*

Specifies that you want to display information ending with records created on or before a specific time on the specified end date using HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that you want to display information ending with records created on or before the current time on the specified end date,

### **NOW**+*numhours:numminutes* or *+numhours:numminutes*

Specifies that you want to display information ending with records created on or before the current time plus a specific number of hours (and, optionally, minutes) on the specified end date.

For example, if you issue the QUERY ACTLOG command at 9:00 with ENDTIME=NOW+3 or ENDTIME=+3, ADSM displays records with a time of 12:00 or later on the specified end date.

### **NOW**-*numhours:numminutes* or *-numhours:numminutes*

Specifies that you want to display information beginning with records created on or before the current time minus a specific number of hours (and, optionally, minutes) on the specified end date.

For example, if you issue the QUERY ACTLOG command at 9:00 with ENDTIME=NOW-3:30 or ENDTIME=-3:30, ADSM displays records with a time of 5:30 or later on the specified end date.

## QUERY ACTLOG

### **MSGno**=*msgnumber*

Specifies an integer that defines the number of the message to be displayed from the activity log. This is the numeric part of the message without the ANR prefix or the message type suffix (that is, I, W, E, S, or D). This parameter is optional. If you do not specify a value for this parameter, all activity log messages are displayed.

### **Search**=*stringexpr*

Specifies a string expression that you want to search for in the activity log. This parameter is optional. The string expression must be enclosed in quotation marks if it contains blanks. You can use a pattern-matching expression to specify this string. All matching strings are displayed. If you do not specify a value for this parameter, all strings are displayed.

## Examples

### Task 1

Search the activity log for all messages appearing on April 27, 1995 (04/27/95), between 14:57 and 15:39.

**Command:** query actlog begindate=04/27/1995 begintime=14:57 endtime=15:39

### Output:

Date/Time	Message
04/27/1995 14:57:30	ANR0402I Session 17 started for administrator TOMC (OS/2).
04/27/1995 15:19:43	ANR0811I Inventory client file expiration started.
04/27/1995 15:19:43	ANR0812I Inventory client file expiration complete: 0 files deleted.
04/27/1995 15:31:30	ANR0482W Session 17 terminated - idle more than 15 minute(s).
04/27/1995 15:38:35	ANR0402I Session 18 started for administrator TOMC (OS/2).

where:

### **Date/Time**

Specifies the date and time that the message was generated.

### **Message**

Specifies the message.

### Task 2

Search the activity log for any message that contains the string "delete." The output includes only messages produced during the past hour.

**Command:** query actlog search=delete

## QUERY ACTLOG

**Output:**

Date/Time	Message
08/27/1995 15:19:43	ANR0812I Inventory client file expiration complete: 0 files deleted.

where:

**Date/Time**

Specifies the date and time that the message was generated.

**Message**

Specifies the message.

**Task 3**

Display messages from the activity log that contain the text string export. Search the activity log for messages that occurred on May 5, 1995 (05/05/1995) after 10:00.

**Command:** query actlog begindate=05/05/1995 begintime=10:00 search=export\*

**Output:**

## QUERY ACTLOG

Date/Time	Message
05-05-1995 10:00:27	ANR1360I Export volume EXP001 opened (sequence number 1).
05-05-1995 10:00:27	ANR0610I EXPORT SERVER started by JANET as process 4.
05-05-1995 10:00:27	ANR0639I EXPORT SERVER: Processing domain ACCDOMAIN.
05-05-1995 10:00:27	ANR0639I EXPORT SERVER: Processing domain STANDARD.
05-05-1995 10:00:27	ANR0640I EXPORT SERVER: Processing policy set ACTIVE in policy domain ACCDOMAIN.
05-05-1995 10:00:27	ANR0640I EXPORT SERVER: Processing policy set STANDARD in policy domain ACCDOMAIN.
05-05-1995 10:00:27	ANR0640I EXPORT SERVER: Processing policy set ACTIVE in policy domain STANDARD.
05-05-1995 10:00:27	ANR0640I EXPORT SERVER: Processing policy set STANDARD in policy domain STANDARD.
05-05-1995 10:00:27	ANR0641I EXPORT SERVER: Processing management class STANDARD in domain ACCDOMAIN, set ACTIVE.
05-05-1995 10:00:27	ANR0641I EXPORT SERVER: Processing management class STANDARD in domain ACCDOMAIN, set STANDARD.
05-05-1995 10:00:27	ANR0641I EXPORT SERVER: Processing management class STANDARD in domain STANDARD, set ACTIVE.
05-05-1995 10:00:27	ANR0641I EXPORT SERVER: Processing management class MC in domain STANDARD, set STANDARD.
05-05-1995 10:00:27	ANR0641I EXPORT SERVER: Processing management class STANDARD in domain STANDARD, set STANDARD.
05-05-1995 10:00:28	ANR0643I EXPORT SERVER: Processing archive copy group in domain ACCDOMAIN, set ACTIVE, management class STANDARD.

where:

### Date/Time

Specifies the date and time that the message was generated.

### Message

Specifies the message.

## Related Commands

Table 90 provides a reference to a related command.

Command	Description	Reference
SET ACTLOGRETENTION	Specifies the number of days to retain log records in the activity log.	Page 507



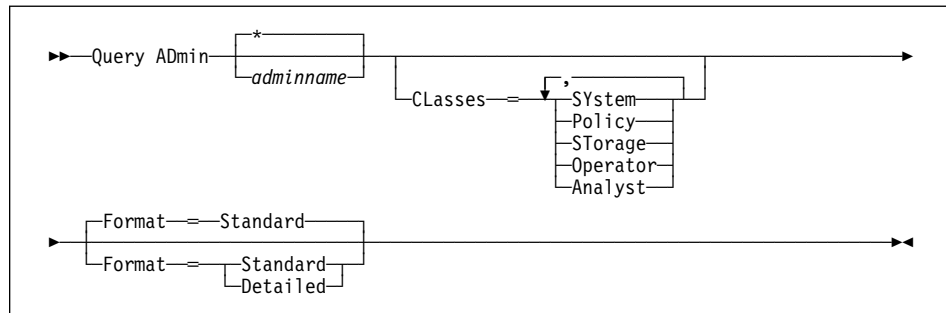
## QUERY ADMIN (Display Information on One or More Administrators)

Use the QUERY ADMIN command to display information on one or more administrators.

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

#### *adminname*

Specifies the name of the administrator for which you want to display information. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching administrator names are included. If you do not specify a value for this parameter, all administrators are displayed.

#### **Classes=***classes*

Specifies that you want to restrict output to those administrators which have been granted at least one of the specific privilege classes. This parameter is optional. You can specify more than one privilege class in a list, with each name delimited with a comma and no intervening spaces. If you do not specify a value for this parameter, information on all specified administrators is displayed, regardless of privilege class. Possible values are:

#### **System**

Specifies that you want to display information on system administrators. A system administrator has the highest level of authority in ADSM. A system administrator can issue any administrative command and has authority to manage all policy domains and all storage pools. Only a system administrator can grant authority to other administrators.

#### **Policy**

Specifies that you want to display information on policy administrators (unrestricted and restricted). A policy administrator can issue commands that relate to policy management objects such as policy domains, policy sets, management classes, copy groups, and schedules.

## QUERY ADMIN

### Storage

Specifies that you want to display information on storage administrators (unrestricted and restricted). A storage administrator can issue commands that allocate and control storage resources for the server.

### Operator

Specifies that you want to display information on operator administrators. An administrator with operator privilege can issue commands that affect the immediate operation of the server, including server availability and physical media.

### Analyst

Specifies that you want to display information on analyst administrators. An administrator with analyst privilege can issue commands that reset the counters which track server performance.

### Format=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed for the specified administrators.

#### Detailed

Specifies that complete information is displayed for the specified administrators.

## Examples

### Task 1

Display partial information on all administrators.

**Command:** query admin

**Output:**

Administrator Name	Days Since Last Access	Days Since Password Set	Locked?	Privilege Classes
ADMIN	<1	<1	No	System
SERVER_CONSOLE			No	System

where:

#### Administrator Name

Specifies the name of the administrator.

#### Days Since Last Access

Specifies the number of days since the administrator last accessed the server.

## QUERY ADMIN

### Days Since Password Set

Specifies the number of days since the administrator's password was defined or most recently updated.

### Locked?

Specifies whether the administrator is locked out of the system.

### Privilege Classes

Specifies the privilege classes for which the administrator has authority.

### Task 2

Display complete information for the administrator named ADMIN.

**Command:** query admin admin format=detailed

### Output:

```
Administrator Name: ADMIN
Last Access Date/Time: 05/31/1996 14:09:18
Days Since Last Access: <1
Password Set Date/Time: 05/31/1996 13:02:56
Days Since Password Set: 1
    Locked?: No
    Contact:
    System Privilege:Yes
    Policy Privilege: **Included with system privilege
    Storage Privilege:**Included with system privilege
    Analyst Privilege:**Included with system privilege
    Operator Privilege:**Included with system privilege
Registration Date/Time:05/31/1996 13:02:56
Registering Administrator:SERVER_CONSOLE
```

where:

### Administrator Name

Specifies the name of the administrator.

### Last Access Date/Time

Specifies the date and time that the administrator last accessed the server.

### Days Since Last Access

Specifies the number of days since the administrator last accessed the server.

### Password Set Date/Time

Specifies the date and time that the administrator's password was defined or most recently updated.

### Days Since Password Set

Specifies the number of days since the administrator's password was defined or most recently updated.

## QUERY ADMIN

**Locked?**

Specifies whether the administrator is locked out of the system.

**Contact**

Specifies any contact information for the administrator.

**System Privilege**

Specifies whether the administrator has been granted system privilege.

**Policy Privilege**

Specifies whether the administrator has been granted unrestricted policy privilege or the names of any policy domains that the restricted policy administrator can manage.

**Storage Privilege**

Specifies whether the administrator has been granted unrestricted storage privilege or the names of any storage pools that the restricted storage administrator can manage.

**Analyst Privilege**

Specifies whether the administrator has been granted analyst privilege.

**Operator Privilege**

Specifies whether the administrator has been granted operator privilege.

**Registration Date/Time**

Specifies the date and time that the administrator was registered.

**Registering Administrator**

Specifies the name of the administrator who registered the administrator.

## QUERY ADMIN

### Related Commands

Table 91 provides references to related commands.

*Table 91. Commands Related to QUERY ADMIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
GRANT AUTHORITY	Assigns one or more privilege classes to an administrator.	Page 273
REGISTER ADMIN	Defines a new administrator without granting administrative authority.	Page 461
REMOVE ADMIN	Removes an administrator from the list of registered administrators.	Page 470
RENAME ADMIN	Changes an ADSM administrator's name.	Page 474
REVOKE AUTHORITY	Revokes one or more privilege classes or restricts access to policy domains and storage pools.	Page 498

## QUERY ASSOCIATION

---

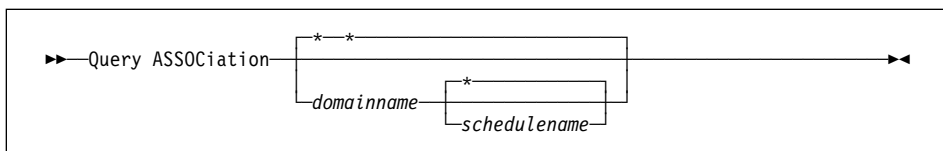
### QUERY ASSOCIATION (Query Client Node Associations with a Schedule)

Use the QUERY ASSOCIATION command to display the client nodes associated with one or more schedules. Client nodes that are associated with a schedule perform backup or archive operations according to that schedule.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domainname*

Specifies the name of the policy domain that you want to display. You can use a pattern-matching expression to specify this name. All matching policy domain names are displayed. If you do not specify a value for this parameter, all existing policy domains are queried. If you specify a domain name, you do not have to specify a schedule name.

##### *schedulingname*

Specifies the name of the schedule that you want to display. You can use a pattern-matching expression to specify this name. All matching schedule names are displayed. If you do not specify a value for this parameter, all existing schedules are queried. If you specify a schedule name, you must also specify a policy domain name.

#### Examples

##### Task 1

Display all the clients that are associated with each schedule that belongs to the EMPLOYEE\_RECORDS policy domain.

**Command:** `query association employee-records *`

##### Output:

```
Policy Domain Name: EMPLOYEE_RECORDS
Schedule Name: WEEKLY_BACKUP
Associated Nodes: JOE JOHNSON LARRY SMITH SMITHERS TOM
```

## QUERY ASSOCIATION

where:

**Policy Domain Name**

Specifies the name of the policy domain to which the schedule belongs.

**Schedule Name**

Specifies the name of the schedule that belongs to the specified policy domain.

**Associated Nodes**

Specifies the names of the clients that are currently associated with the specified schedule.

**Task 2**

Display all the clients that are assigned to the DOMAIN1 policy domain and associated with the WINTER schedule.

**Command:** `query association domain1 winter`

**Output:**

```
Policy Domain Name: DOMAIN1
Schedule Name: WINTER
Associated Nodes: DAVE DAVID LARRY MARIE
```

where:

**Policy Domain Name**

Specifies the name of the policy domain to which the schedule belongs.

**Schedule Name**

Specifies the name of the schedule that belongs to the specified policy domain.

**Associated Nodes**

Specifies the names of the clients that are currently associated with the specified schedule.

## QUERY ASSOCIATION

### Related Commands

Table 92 provides references to related commands.

*Table 92. Commands Related to QUERY ASSOCIATION*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE ASSOCIATION	Associates one or more clients with a schedule.	Page 100
DELETE ASSOCIATION	Deletes the association between one or more clients and a schedule.	Page 197



## QUERY AUDITOCUPANCY

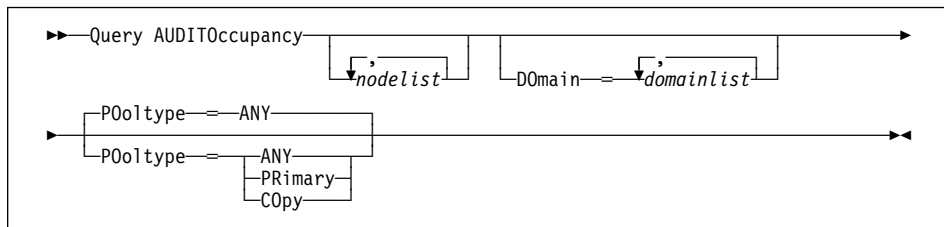
### QUERY AUDITOCUPANCY (Query Client Node Storage Utilization)

Use the QUERY AUDITOCUPANCY command to display information about the client node data storage utilization. The displayed information is current as of the last license audit processed by the ADSM server. You can use this information to determine if and where client node storage utilization needs to be balanced. This information can also assist with billing clients for storage usage.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *nodelist*

Specifies a list of nodes for which data storage utilization information is to be displayed. You can specify more than one node by separating the node names with commas, with no intervening spaces. This parameter is optional. You can use pattern-matching expressions to specify names. The default (\*) is to query all client nodes. You can also limit this list by policy domain via the optional DOMAIN parameter.

##### **DOMAIN=domainlist**

Specifies a list of policy domains to restrict which nodes are displayed. This parameter is optional. Only nodes belonging to the specified policy domains are displayed. You can specify more than one policy domain by separating the policy domain names with commas, with no intervening spaces. You can use pattern-matching expressions to specify names.

##### **POOLTYPE=value**

Specifies whether to display storage usage in primary storage pools only, copy storage pools only, or both primary and copy storage pools. This parameter is optional. The default is ANY. Possible values are:

##### **ANY**

Specifies that the storage usage displayed includes files stored in both primary and copy storage pools. The value presented is the total for the two pools.

##### **Primary**

Specifies that storage usage in primary storage pools is displayed.

## QUERY AUDITOCUPANCY

### COPY

Specifies that storage usage in copy storage pools is displayed.

## Examples

### Task

Display combined storage utilization in primary and copy storage pools.

**Command:** query auditoccupancy

### Output:

```
License information as of last audit on 05/22/1996 14:49:51.
```

Client Node Name	Backup Storage Used (MB)	Archive Storage Used (MB)	Space-Managed Storage Used (MB)	Total Storage Used (MB)
CLIENT	245	20	0	265
SMITH	245	20	0	265
SMITHERS	245	20	0	265
JOHNSON	300	15	0	320
JOE	245	20	0	265
TOM	300	15	0	320
LARRY	245	20	0	265

where:

### Client Node Name

Specifies the name of the client node.

### Backup Storage Used (MB)

Specifies the total backup storage utilization for the node.

### Archive Storage Used (MB)

Specifies the total archive storage utilization for the node.

### Space-Managed Storage Used (MB)

Specifies the amount of server storage used to store files migrated from the client node.

### Total Storage Used (MB)

Specifies the total client storage utilization (for backup, archive, and space management) for the node.

## QUERY AUDITOCUPANCY

### Related Commands

Table 93 provides references to related commands.

*Table 93. Commands Related to QUERY AUDITOCUPANCY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LICENSES	Checks for compliance with defined licenses.	Page 54
QUERY LICENSE	Displays information about licenses and audits.	Page 386
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
REGISTER LICENSE	Registers a new license with the ADSM server.	Page 463
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.	Page 512

## QUERY CONTENT

### QUERY CONTENT (Query the Contents of a Storage Pool Volume)

Use the QUERY CONTENT command to display information about one or more files currently residing in a storage pool volume. This command can also be used to identify files that have been found to be damaged and files that have been backed up to a copy storage pool.

This command is useful when a volume is damaged or before you:

- Request the server to fix any inconsistencies between a volume and the database
- Move files from one volume to another volume
- Delete a volume from a storage pool

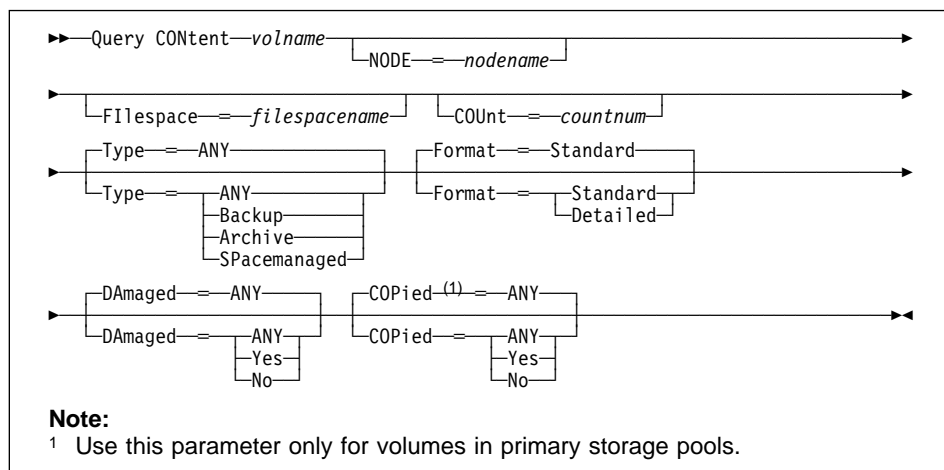
Because the results of this command can take a long time to execute and can be extremely large, it is recommended that you use the COUNT parameter to limit the number of files displayed.

**Note:** Files in cache that are marked as damaged are never displayed.

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

*volname*

Specifies the name of the volume to be queried. This parameter is required.

**NODE**=*nodename*

Specifies the name of the backup-archive client or space management client associated with the file space to be queried. This parameter is optional. You

## QUERY CONTENT

can use a pattern-matching expression to specify this name. If you do not specify a name, all backup-archive and space management clients are included.

**Filespace**=*filespace**name*

Specifies the name of the file space to queried. This parameter is optional. You can use a pattern-matching expression to specify this name. File space names are case-sensitive. If you do not specify a file space name, all file spaces are included.

**COUnT**=*countnum*

Specifies the number of files to be displayed. This parameter is optional. You can specify either a negative integer or a positive integer. If you specify a positive integer, *n*, the first *n* files are displayed. If you specify a negative integer, *-n*, the last *n* files are displayed in *reverse* order. You cannot specify COUNT=0. If you do not specify a value for this parameter, all files are displayed.

**Type**=*filetype*

Specifies the type of file space to be queried. This parameter is optional. The default value is ANY. Possible values are:

**ANY**

Specifies that all types of files in the specified storage pool volume are queried: backup versions of files, archived copies of files, and files migrated from one or more client nodes.

**Backup**

Specifies that only backup file spaces are queried.

**Archive**

Specifies that only archive file spaces are queried.

**SPacemanaged**

Specifies that only space management file spaces (files migrated from one or more client nodes) are queried.

**Format**=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

**Standard**

Specifies that partial information is displayed for the specified storage pool volume.

**Detailed**

Specifies that complete information is displayed for the specified storage pool volume.

**DAMaged**=*value*

Specifies whether to restrict the query output either to files that are known to be damaged, or to files that are not known to be damaged. This parameter is optional. The default value is ANY.

## QUERY CONTENT

Possible values are:

### **ANY**

Specifies that files are displayed regardless of whether the files have been found to be damaged.

### **Yes**

Specifies that only files that have been damaged are displayed. These are files that were found to have data integrity errors when a user attempted to restore, retrieve, or recall the file, or when an AUDIT VOLUME operation was run.

### **No**

Specifies that only files *not* known to have data integrity errors are displayed.

### **COPIED=value**

Specifies whether to restrict the query output either to files that are backed up to a copy storage pool, or to files that are not backed up to a copy storage pool. This parameter is optional. The default value is ANY. Possible values are:

### **ANY**

Specifies that files are displayed regardless of whether the files are backed up to a copy storage pool. Primary and cached file copies are displayed.

### **Yes**

Specifies that the only files displayed are those for which at least one, usable backup copy exists in a copy storage pool. A file is not displayed if its copy in the copy storage pool is known to have data integrity errors. Cached file copies are not displayed, because these files are never restored.

Use COPIED=YES to identify primary files that can be restored using the RESTORE VOLUME or RESTORE STGPOOL command.

### **No**

Specifies that the only files displayed are those for which no usable backup copies exist in a copy storage pool. Cached file copies are not displayed, because these files are never restored.

Use COPIED=NO to identify primary files that cannot be restored using the RESTORE VOLUME or RESTORE STGPOOL command.

## Examples

### **Task 1**

Query the contents of the volume named PAYROLL.STG1. Only display the first four files of the output.

**Command:** query content payroll.stg1 count=4

## QUERY CONTENT

### Output:

Node Name	Type	Filespace Name	Client's Name for File
BIGBUCKS	Bkup	OS2C	\ AUTOEXEC.BAT
BIGBUCKS	Bkup	OS2C	\ CONFIG.SYS
BIGBUCKS	Bkup	OS2C	\ MRFILEPM.DBG
BIGBUCKS	Bkup	OS2C	\ OS2KRNL

where:

#### **Node Name**

Specifies the name of the node to which the specified file belongs.

#### **Type**

Specifies the type of file, either BACKUP or ARCHIVE.

#### **Filespace Name**

Specifies the name of the file space to which the specified file belongs.

#### **Client's Name for File**

Specifies the name of the file.

### Task 2

Query the contents of the volume named STGVOL.1. Only display the last 3 files of the output.

**Command:** `query content stgvol.1 count=-3 format=detailed`

## QUERY CONTENT

### Output:

```
Node Name: BILL
Type: Bkup
Filespace Name: DDRIVE
Client's Name for File: \APPS\TRASHCAN\TOOLKIT\ASM\
                        OS2INC\ VLPT.INC
Stored Size: 435
Segment Number: 1/1
Cached Copy?: Yes

Node Name: BILL
Type: Bkup
Filespace Name: DDRIVE
Client's Name for File: \APPS\TRASHCAN\TOOLKIT\ASM\
                        OS2INC\ VKBD.INC
Stored Size: 2,358
Segment Number: 1/1
Cached Copy?: Yes

Node Name: BILL
Type: Bkup
Filespace Name: DDRIVE
Client's Name for File: \APPS\TRASHCAN\TOOLKIT\ASM\
                        OS2INC\ VDOS.INC
Stored Size: 962
Segment Number: 1/1
Cached Copy?: Yes
```

where:

#### **Node Name**

Specifies the name of the node to which the specified file belongs.

#### **Type**

Specifies the type of file, either BACKUP or ARCHIVE.

#### **Filespace Name**

Specifies the name of the file space to which the specified file belongs.

#### **Client's Name for File**

Specifies the name of the file.

#### **Stored Size**

Specifies the size of the file, in bytes.

#### **Segment Number**

Specifies whether the file is stored across multiple volumes. For example, if the specified file is stored across two volumes, the segment number indicates *1/2* (the first part of the file is stored on the specified volume) or *2/2* (the second part



## QUERY CONTENT

of the file is stored on the specified volume). If the segment number is 1/1, the file is completely stored on the specified volume.

### Cached Copy?

Specifies whether the specified file is a cached copy of a migrated file.

## Related Commands

Table 94 provides references to related commands.

*Table 94. Commands Related to QUERY CONTENT*

Command	Description	Reference
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
RESTORE STGPOOL	Restores files to a primary storage pool from copy storage pools.	Page 488
RESTORE VOLUME	Restores files stored on specified volumes in a primary storage pool from copy storage pools.	Page 493
UPDATE VOLUME	Updates the attributes of one or more storage pool volumes.	Page 619

## QUERY COPYGROUP

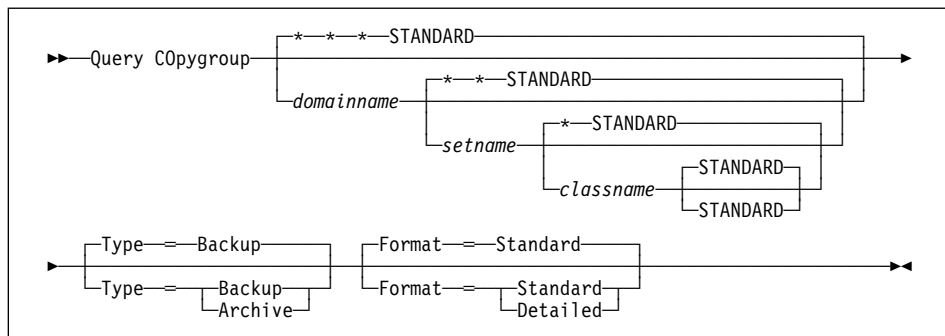
### QUERY COPYGROUP (Query One or More Copy Groups)

Use the QUERY COPYGROUP command to display information about one or more copy groups.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domainname*

Specifies the name of the policy domain associated with the copy group to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy domains are queried. If you do not specify a value for this parameter, all policy domains are queried. You must specify this parameter when querying an explicitly named copy group.

##### *setname*

Specifies the name of the policy set associated with the copy group to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy sets are queried. If you do not specify a value for this parameter, all policy sets are queried. You must specify this parameter when querying an explicitly named copy group.

##### *classname*

Specifies the name of the management class associated with the copy group to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching management classes are queried. If you do not specify a value for this parameter, all management classes are queried. You must specify this parameter when querying an explicitly named copy group.

##### **STANDARD**

Specifies the name of the copy group. This parameter is optional. The name of the copy group must be STANDARD. If you do not specify a value for this parameter, the system defaults to STANDARD.

## QUERY COPYGROUP

**Type=***type*

Specifies the type of copy group to be queried. This parameter is optional. The default value is BACKUP. Possible values are:

**Backup**

Specifies that you want to query backup copy groups.

**Archive**

Specifies that you want to query archive copy groups.

**Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

**Standard**

Specifies that partial information is displayed for the specified copy group.

**Detailed**

Specifies that complete information is displayed for the specified copy group.

## Examples

### Task 1

Display complete information on the backup copy group assigned to the ACTIVEFILES management class in the VACATION policy set of the EMPLOYEE\_RECORDS policy domain.

**Command:** query copygroup employee\_records vacation activefiles  
format=detailed

**Output:**

```
Policy Domain Name: EMPLOYEE_RECORDS
Policy Set Name: VACATION
Mgmt Class Name: ACTIVEFILES
Copy Group Name: STANDARD
Copy Group Type: Backup
Versions Data Exists: 5
Versions Data Deleted: 1
Retain Extra Versions: 30
Retain Only Version: 60
Copy Mode: Absolute
Copy Serialization: Shared Static
Copy Frequency: 3
Copy Destination: BACKUPPOOL
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 13:15:47
```

## QUERY COPYGROUP

where:

**Policy Domain Name**

Specifies the name of the policy domain.

**Policy Set Name**

Specifies the name of the policy set.

**Mgmt Class Name**

Specifies the name of the management class.

**Copy Group Name**

Specifies the name of the copy group. This name must be STANDARD.

**Copy Group Type**

Specifies the type of the copy group.

**Versions Data Exists**

Specifies the maximum number of backup versions that are retained by the server for files that are on the client's file system.

**Versions Data Deleted**

Specifies the maximum number of backup versions that are retained by the server for files that have been deleted from the client's file system.

**Retain Extra Versions**

Specifies the retention time, in days, for all but the most recent backup version.

**Retain Only Version**

Specifies the retention time, in days, for the last backup version of a file that has been deleted from the client's file system.

**Copy Mode**

Specifies whether files in the copy group are backed up depending on whether the file has been modified since the last backup. The copy mode value is only used for incremental backup, this value is ignored during selective backup. This value can be MODIFIED or ABSOLUTE.

**Copy Serialization**

Specifies whether a file can be in use during a backup operation.

**Copy Frequency**

Specifies the minimum interval in days between backups.

**Copy Destination**

Specifies the name of the storage pool where the backup copy group's files are to reside.

**Last Update by (administrator)**

Specifies the name of the administrator that defined or most recently updated the copy group.

## QUERY COPYGROUP

### Last Update Date/Time

Specifies the date and time that the administrator defined or most recently updated the copy group.

### Task 2

Display complete information on the archive copy group STANDARD that is assigned to the ACTIVEFILES management class in the VACATION policy set of the EMPLOYEE\_RECORDS policy domain.

**Command:** query copygroup employee\_records vacation activefiles  
type=archive format=detailed

### Output:

```
Policy Domain Name: EMPLOYEE_RECORDS
Policy Set Name: VACATION
Mgmt Class Name: ACTIVEFILES
Copy Group Name: STANDARD
Copy Group Type: Archive
Retain Version: 730
Copy Serialization: Shared Static
Copy Frequency: CMD
Copy Mode: ABSOLUTE
Copy Destination: ARCHIVEPOOL
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 13:15:47
```

where:

### Policy Domain Name

Specifies the name of the policy domain.

### Policy Set Name

Specifies the name of the policy set.

### Mgmt Class Name

Specifies the name of the management class.

### Copy Group Name

Specifies the name of the copy group. This name must be STANDARD.

### Copy Group Type

Specifies the type of the copy group.

### Retain Version

Specifies the number of days to keep an archive copy.

### Copy Serialization

Specifies whether a file can be in use during an archive operation.

## QUERY COPYGROUP

### Copy Frequency

Specifies the copy frequency of the copy group. For archive copy groups, this value must be CMD.

### Copy Mode

Specifies that files in the copy group are archived regardless of whether they have been modified. For archive copy groups, this value must be ABSOLUTE.

### Copy Destination

Specifies the name of the storage pool where the archive copy group's files are to reside.

### Last Update by (administrator)

Specifies the name of the administrator that defined or most recently updated the copy group.

### Last Update Date/Time

Specifies the date and time that the administrator defined or most recently updated the copy group.

## Related Commands

Table 95 provides references to related commands.

*Table 95. Commands Related to QUERY COPYGROUP*

Command	Description	Reference
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DELETE COPYGROUP	Deletes a backup or archive copy group from a policy domain and policy set.	Page 199
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539

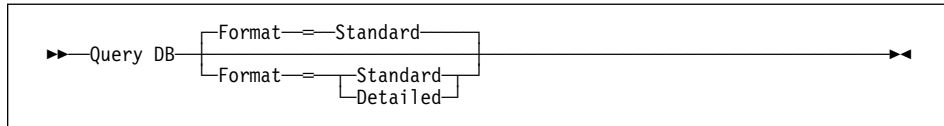
## QUERY DB (Display Information on the Database)

Use the QUERY DB command to display allocation information for the database, including information about utilization, expansion, and reduction abilities.

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

#### Format=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed for the database.

#### Detailed

Specifies that complete information is displayed for the database.

### Examples

#### Task 1

Display statistical information about the database.

**Command:** query db

**Output:**

Available Space (MB)	Assigned Capacity (MB)	Maximum Extension (MB)	Maximum Reduction (MB)	Page Size (bytes)	Total Pages	Used Pages	%Util	Max. %Util
8	4	4	0	4,096	1,024	94	9.2	9.2

where:

#### Available Space (MB)

Specifies the amount of space available in the database, in megabytes.

## QUERY DB

### **Assigned Capacity (MB)**

Specifies the assigned storage capacity for the database, in megabytes.

### **Maximum Extension (MB)**

Specifies the maximum amount the database can be extended, in megabytes.

### **Maximum Reduction (MB)**

Specifies the maximum amount the database can be reduced, in megabytes.

### **Page Size (bytes)**

Specifies the page size, in bytes.

### **Total Pages**

Specifies the total number of pages available in the database. A page is a unit of allocation within database volumes. The total number of pages will always be a multiple of the Assigned Capacity value, by a factor of 256 (1 megabyte/256 pages).

### **Used Pages**

Specifies the total number of pages currently being used by the database.

### **%Util**

Specifies the current utilization value as a percentage of the value for the assigned capacity.

### **Max. %Util**

Specifies the maximum occupancy for the database, as a percentage of the available space. This value specifies the highest capacity occupied, or "high water mark" for database utilization. You can reset this value with the RESET DBMAXUTILIZATION command.

### **Task 2**

Display statistical information about the database in complete detail.

**Command:** `query db format=detailed`



Output:

```

    Available Space (MB): 8
    Assigned Capacity (MB): 4
    Maximum Extension (MB): 4
    Maximum Reduction (MB): 0
    Page Size (bytes): 4,096
    Total Usable Pages: 1,024
        Used Pages: 94
            %Util: 9.2
        Max. %Util: 9.2
    Physical Volumes: 3
    Buffer Pool Pages: 64
    Total Buffer Requests: 4,702
    Cache Hit Pct.: 97.68
    Cache Wait Pct.: 0.00
    Backup in Progress?: No
    Type of Backup In Progress:
    Incrementals Since Last Full: 0
    Changed Since Last Backup (MB): 0.37
    Percentage Changed: 100.00
    Last Complete Backup Date/Time:
  
```

where:

**Available Space (MB)**

Specifies the amount of space available in the database, in megabytes.

**Assigned Capacity (MB)**

Specifies the assigned storage capacity for the database, in megabytes.

**Maximum Extension (MB)**

Specifies the maximum amount the database can be extended, in megabytes.

**Maximum Reduction (MB)**

Specifies the maximum amount the database can be reduced, in megabytes.

**Page Size (bytes)**

Specifies the page size, in bytes.

**Total Usable Pages**

Specifies the total number of pages available in the database. A page is a unit of allocation within database volumes. The total number of pages will always be a multiple of the Assigned Capacity value, by a factor of 256 (1 megabyte/256 pages).

## QUERY DB

### **Used Pages**

Specifies the total number of pages currently being used by the database.

### **%Util**

Specifies the current utilization as a percentage value of the assigned capacity.

### **Max. %Util**

Specifies the maximum occupancy for the database, as a percentage of the available space. This value specifies the highest capacity occupied, or "high water mark" for database utilization. You can reset this value with the RESET DBMAXUTILIZATION command.

### **Changed Pages**

Specifies the total number of pages changed since the last backup of the database.

### **Physical Volumes**

Specifies the number of volumes defined to the database, including database copies.

### **Buffer Pool Pages**

Specifies the number of pages in the database buffer pool. This value is determined by the value for the BUFFPOOLSIZ option. The default value for the BUFFPOOLSIZ option is 512 kilobytes (128 pages).

### **Total Buffer Requests**

Specifies the cumulative number of requests for database pages, since the last RESET BUFPOOL command (or since the server was last started). You can reset this value with the RESET BUFPOOL command.

### **Cache Hit Pct.**

Specifies, as a percentage of the total number of database requests, the number of requests for a database page that is already in the database buffer pool. Requesting pages that are already in the buffer pool increases database performance. You can reset this value with the RESET BUFPOOL command.

### **Cache Wait Pct.**

Specifies, as a percentage, the number requests for a database buffer pool page that was unavailable (because all database buffer pool pages are occupied). If this number is greater than 0.0, you should increase the size of the buffer pool with the

## QUERY DB

BUFFPOOLSIZE option. You can reset this value with the RESET BUFPOOL command.

### Backup in Progress?

Specifies whether a backup of the database is in progress.

### Type of Backup in Progress

Specifies whether the type of backup of the database is a full or incremental backup.

### Incremental Since Last Full

Specifies the number of incremental backups that have been run since the last full backup of the database was done.

### Changed Since Last Backup (MB)

Specifies the database pages (in megabytes) that have changed since the last backup of the database.

### Percentage Changed

Specifies, as a percentage, the number of database pages that changed since the last backup of the database.

### Last Complete Backup Date/Time

Specifies the data and time of the last complete backup of the database.

## Related Commands

Table 96 provides references to related commands.

Table 96 (Page 1 of 2). Commands Related to QUERY DB

Command	Description	Reference
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
EXTEND DB	Extends usable database storage within the volumes previously allocated for its use.	Page 269

## QUERY DB

*Table 96 (Page 2 of 2). Commands Related to QUERY DB*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
EXTEND LOG	Extends usable log storage within the volumes previously allocated for its use.	Page 271
REDUCE DB	Attempts to free up a database volume by reducing usable storage within the volumes defined.	Page 457
REDUCE LOG	Attempts to free up a log volume by reducing usable storage within the volumes defined.	Page 459
RESET BUFPOOL	Resets the buffer pool statistics for the database.	Page 482
RESET DBMAXUTILIZATION	Resets the maximum utilization statistic for the database.	Page 484

## QUERY DBBACKUPTRIGGER

---

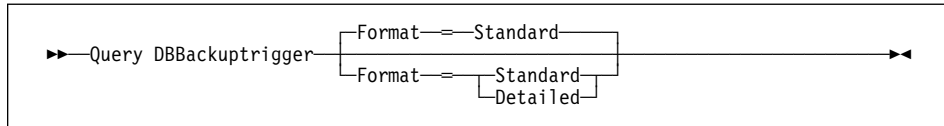
### QUERY DBBACKUPTRIGGER (Query the Database Backup Trigger)

Use the QUERY DBBACKUPTRIGGER command to display the current settings for the database backup trigger.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the database backup trigger.

##### **Detailed**

Specifies that complete information is displayed for the database backup trigger.

#### Examples

##### Task

Display the current settings for the database backup trigger.

**Command:** query dbbackuptrigger format=detailed

##### Output:

```
Full Device Class: FILE
Incremental Device Class: FILE
Log Full Percentage: 50
Incrementals Between Fulls: 6
Last Update by (administrator): JANET
Last Update Date/Time: 05/31/1995 11:15:35
```

where:

##### **Full Device Class**

Specifies the name of the device class used for full backups.

## QUERY DBBACKUPTRIGGER

### Incremental Device Class

Specifies the name of the device class used for incremental backups.

### Log Full Percentage

Specifies the percentage of space used for the recovery log at which ADSM performs an automatic backup of the database.

### Incrementals Between Fulls

Specifies the number of incremental backups ADSM runs between full backups of the database.

### Last Update by (administrator)

Specifies the name of the administrator that most recently updated the database.

### Last Update Date/Time

Specifies when the administrator most recently updated the database.

## Related Commands

Table 97 provides references to related commands.

Command	Description	Reference
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
DELETE DBBACKUPTRIGGER	Deletes the database backup trigger.	Page 202
UPDATE DBBACKUPTRIGGER	Changes one or more attributes of the database backup trigger.	Page 547

## QUERY DBVOLUME

---

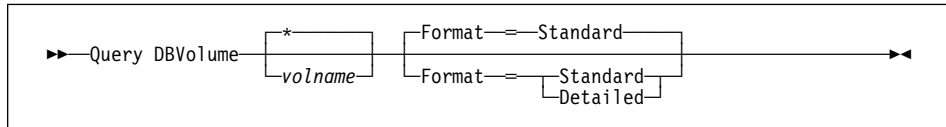
### QUERY DBVOLUME (Display Information on One or More Database Volumes)

Use the QUERY DBVOLUME command to display information on one or more database volumes, including available, allocated, and free space on the volume. This command displays information about the specified database volume and any database volume copies.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *volname*

Specifies the name of the database volume to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching volume names are displayed. If you do not specify a volume name, information on all database volumes is displayed.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information for the specified database volumes is displayed.

##### **Detailed**

Specifies that complete information for the specified database volumes is displayed.

#### Examples

##### Task 1

Display information on the database volume named db.1, located on the D drive in the DBVOL directory.

**Command:** query dbvolume d:\dbvol\db.1

**Output:**

## QUERY DBVOLUME

Volume Name (Copy 1)	Copy Status	Volume Name (Copy 2)	Copy Status	Volume Name (Copy 3)	Copy Status
D:\dbvol\db.1	Sync'd		Undef- ined		Undef- ined

where:

### Volume Name (Copy *n*)

Specifies the name of the database volume. The output displays status of the three volume copies (Copy 1, Copy 2, and Copy 3).

### Copy Status

Specifies the status of the database volume. Possible values are:

#### Offline

The volume is varied offline.

#### Stale

The volume is not synchronized with the other volumes in the database and cannot be used for storage.

#### Sync'd

The volume is synchronized with the other volumes in the database and can be used for storage.

#### Undefined

No volume has been defined.

## Task 2

Display information in complete detail on the database volume named db.1, located on the D drive in the DBVOL directory.

**Command:** `query dbvolume d:\dbvol\db.1 format=detailed`

### Output:

```
Volume Name (Copy 1): d:\dbvol\db.1
    Copy Status: Sync'd
Volume Name (Copy 2):
    Copy Status: Undefined
Volume Name (Copy 3):
    Copy Status: Undefined
Available Space (MB): 4
Allocated Space (MB): 4
    Free Space (MB): 0
```



## QUERY DBVOLUME

where:

### Volume Name (Copy *n*)

Specifies the name of the database volume. The output displays status of the three volume copies (Copy 1, Copy 2, and Copy 3).

### Copy Status

Specifies the status of the database volume. Possible values are:

#### Offline

The volume is varied offline.

#### Stale

The volume is not synchronized with the other volumes in the database and cannot be used for storage.

#### Sync'd

The volume is synchronized with the other volumes in the database and can be used for storage.

#### Undefined

No volume has been defined.

### Available Space (MB)

Specifies the total amount of space that is available, both allocated and free, for the database volume, in megabytes.

### Allocated Space

Specifies the amount of space allocated to the database volume, in megabytes.

### Free Space

Specifies the amount of unassigned space for this mirror set that can be allocated for use, in megabytes.

## Related Commands

Table 98 provides references to related commands.

Table 98. Commands Related to QUERY DBVOLUME

Command	Description	Reference
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
DELETE DBVOLUME	Deletes a volume from the database.	Page 204
VARY	Specifies whether a disk volume is available to the server for use.	Page 629

## QUERY DEVCLASS

---

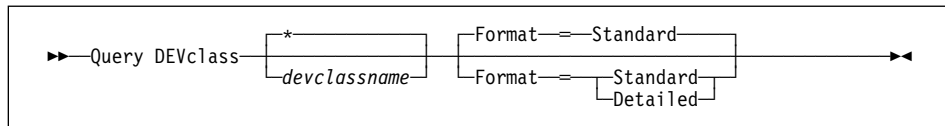
### QUERY DEVCLASS (Display Information on One or More Device Classes)

Use the QUERY DEVCLASS command to display information on one or more device classes.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *devclassname*

Specifies the name of the device class to be queried. This parameter is optional. You can use wildcard characters to specify this name. All matching device classes are displayed. If you do not specify a value for this parameter, all device classes are displayed.

##### **Format**=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the specified device class.

##### **Detailed**

Specifies that complete information is displayed for the specified device class.

#### Examples

##### Task 1

Display information on all device classes.

**Command:** query devclass

## QUERY DEVCLASS

### Output:

Device Class Name	Device Access Strategy	Storage Pool Count	Device Type	Format	Est/Max Capacity (MB)	Mount Limit
8MMTAPE	Sequential	1	8MM	DRIVE	6,144.0	2
DISK	Random	4				
PLAINFILES	Sequential	1	FILE		50.0	1

where:

#### Device Class Name

Specifies the name of the device class.

#### Device Access Strategy

Specifies how data is written to the device class.

#### Storage Pool Count

Specifies the number of storage pools that are assigned to the device class.

#### Device Type

Specifies the device type of the device class.

#### Format

Specifies the recording format.

#### Est/Max Capacity (MB)

Specifies the estimated or maximum capacity of a volume associated with this device class.

#### Mount Limit

Specifies the maximum number of sequential access volumes that can be mounted concurrently.

### Task 2

Display information in full detail on the PLAINFILES device class.

**Command:** `query devclass plainfiles format=detailed`

## QUERY DEVCLASS

### Output:

```
Device Class Name: PLAINFILES
Device Access Strategy: Sequential
Storage Pool Count: 1
Device Type: FILE
Format:
Est/Max Capacity (MB): 50.0
Mount Limit: 1
Mount Wait (min):
Mount Retention (min):
Label Prefix:
Drive Letter:
Library:
Directory: d:\adsmv2
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 13:15:36
```

where:

#### **Device Class Name**

Specifies the name of the device class.

#### **Device Access Strategy**

Specifies how data is written to the device class.

#### **Storage Pool Count**

Specifies the number of storage pools that are assigned to the device class.

#### **Device Type**

Specifies the device type of the device class.

#### **Format**

Specifies the recording format.

#### **Est/Max Capacity (MB)**

Specifies the estimated or maximum capacity of a volume associated with this device class.

#### **Mount Limit**

Specifies the maximum number of sequential access volumes that can be mounted concurrently.

#### **Mount Wait (min)**

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted.

#### **Mount Retention (min)**

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it.

## QUERY DEVCLASS

### Label Prefix

Specifies the high-level qualifier of the data set name which the server writes into the sequential access media labels.

### Drive Letter

Specifies the drive letter, for example, d for optical devices. server writes into the sequential access media labels.

### Library

Specifies the name of the defined library object that contains the drives used by this device class.

### Directory

Specifies, when applicable, the directory location of the file or files used in a device class.

### Last Update by (administrator)

Specifies the administrator that made the last update to the device class.

### Last Update Date/Time

Specifies the date and time of the last update.

## Related Commands

Table 99 provides references to related commands.

*Table 99. Commands Related to QUERY DEVCLASS*

Command	Description	Reference
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DELETE DEVCLASS	Deletes a device class name.	Page 206
UPDATE DEVCLASS	Changes the attributes of a device class.	Page 550

## QUERY DOMAIN

---

### QUERY DOMAIN (Query One or More Policy Domains)

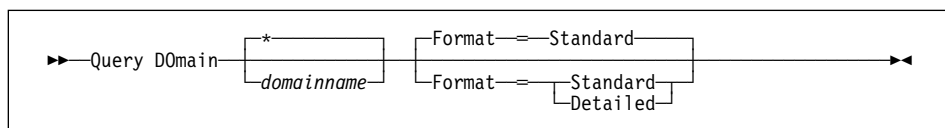
Use the QUERY DOMAIN command to display information on one or more policy domains. The QUERY DOMAIN command is useful in displaying the name of the policy set last activated and its default management class.

There is a difference between the “activated policy set,” as displayed in the QUERY DOMAIN command and the “ACTIVE” policy set. When a policy set is activated, the contents of the policy set are copied to a policy set that has the reserved name ACTIVE. Once activated, there is no real relationship between the policy set that has been activated (copied to ACTIVE) and the contents of the ACTIVE policy set. The original policy set can still be modified, but the copied definitions in the ACTIVE policy set can only be modified by activating another policy set. Because of this copying process, you should not assume that the definitions in the policy set that was last activated are the same as those in the ACTIVE policy set; the original policy set may have been modified since activation.

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

#### *domainname*

Specifies the name of the policy domain to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy domains are displayed. If you do not specify a value for this parameter, all policy domains are displayed.

#### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the specified policy domain.

##### **Detailed**

Specifies that complete information is displayed for the specified policy domain.

## QUERY DOMAIN

### Examples

#### Task 1

Display partial information for all policy domains.

**Command:** query domain

## QUERY DOMAIN

### Output:

Policy Domain Name	Activated Policy Set	Activated Default Mgmt Class	Number of Registered Nodes	Description
EMPLOYEE-RECORDS	VACATION	ACTIVEFI-LES	6	
PROG1			0	Programming Group Domain
PROG2			0	Programming Group Domain
STANDARD	STANDARD	STANDARD	1	Installed default policy domain

where:

#### Policy Domain Name

Specifies the name of the policy domain.

#### Activated Policy Set

Specifies the name of the policy set that was last activated in the domain.

#### Activated Default Mgmt Class

Specifies the assigned default management class for the policy set.

#### Number of Registered Nodes

Specifies the number of client nodes registered to the policy domain.

#### Description

Specifies a description of the policy domain.

### Task 2

Display information in full detail for the EMPLOYEE\_RECORDS policy domain.

**Command:** `query domain employee_records format=detailed`

### Output:



## QUERY DOMAIN

```
Policy Domain Name: EMPLOYEE_RECORDS
Activated Policy Set: VACATION
Activation Date/Time: 05/31/1996 13:15:52
Days Since Activation: <1
Activated Default Mgmt Class: ACTIVEFILES
Number of Registered Nodes: 6
Description:
Backup Retention (Grace Period): 30
Archive Retention (Grace Period): 365
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 13:15:52
```

where:

**Policy Domain Name**

Specifies the name of the policy domain.

**Activated Policy Set**

Specifies the name of the policy set that was last activated in the domain.

**Activation Date/Time**

Specifies the date and time that the policy set was activated.

**Days Since Activation**

Specifies the number of days since the policy set was activated.

**Activated Default Mgmt Class**

Specifies the assigned default management class for the policy set.

**Number of Registered Nodes**

Specifies the number of client nodes registered to the policy domain.

**Description**

Specifies a description of the policy domain.

**Backup Retention (Grace Period)**

Specifies the number of days to retain inactive backup versions of files when any of the following conditions are present:

- A file's management class is changed, but neither the new management class nor default management class contains a backup copy group.
- The management class to which a file is bound no longer exists, and the default management class does not contain a backup copy group.
- The backup copy group is deleted from the management class to which a file is bound and the default management class does not contain a backup copy group.

## QUERY DOMAIN

### Archive Retention (Grace Period)

Specifies the number of days to retain an archive file that meets either of the following conditions:

- The management class to which a file is bound no longer exists, and the default management class does not contain an archive copy group.
- The archive copy group is deleted from the management class to which a file is bound and the default management class does not contain an archive copy group.

### Last Update by (administrator)

Specifies the name of the administrator that defined or most recently updated the policy domain.

### Last Update Date/Time

Specifies when the administrator defined or most recently updated the storage pool.

## Related Commands

Table 100 provides references to related commands.

*Table 100. Commands Related to QUERY DOMAIN*

Command	Description	Reference
COPY DOMAIN	Creates a copy of an existing policy domain.	Page 89
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144
DELETE DOMAIN	Deletes a policy domain along with any policy objects in the policy domain.	Page 208
UPDATE DOMAIN	Changes the attributes of a policy domain.	Page 573

## QUERY DRIVE

---

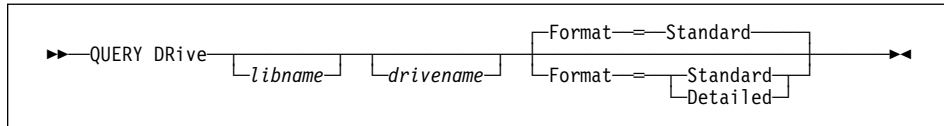
### QUERY DRIVE (Display Information about a Drive)

Use the QUERY DRIVE command to display information about a drive located in a server-attached library.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *libname*

Specifies the name of the library where the queried drive is located. This parameter is optional. You can use a pattern-matching expression to specify this name.

You must specify a value for this parameter if you specify DRIVENAME.

##### *drivename*

Specifies the name assigned to the drive. This parameter is optional. You can use a pattern-matching expression to specify this name.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the drive.

##### **Detailed**

Specifies that complete information is displayed for the drive.

#### Examples

##### Task 1

Display information about all drives associated with your server.

**Command:** query drive

## QUERY DRIVE

### Output:

Library Name	Drive Name	Device Type	Device	Element
AUTO	8MM.0	8MM	tape01\$	116
AUTO	8MM.2	8MM	tape02\$	118
MANUAL	8MM.3	8MM	tape00\$	

where:

#### Library Name

Specifies the name of the library to which the drive is assigned.

#### Drive Name

Specifies the name assigned to the drive.

#### Device Type

Specifies the type of device.

#### Device

Specifies the name of the device that corresponds to the drive.

#### Element

Specifies the element address of the drive.

### Task 2

Display information in full detail about the 8mm drive named DRIVE1 located in the library named AUTO.

**Command:** query drive auto drive1 format=detailed

### Output:

```
Library Name: AUTO
Drive Name: DRIVE1
Device Type: 8MM
Device: tape01$
Element: 116
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 1995-01-13 15:25:03
```

where:

#### Library Name

Specifies the name of the library to which the drive is assigned.

## QUERY DRIVE

**Drive Name**

Specifies the name assigned to the drive.

**Device Type**

Specifies the type of device.

**Device**

Specifies the name of the device that corresponds to the drive.

**Element**

Specifies the element address of the drive.

**Last Update by (administrator)**

Specifies who performed the last update to the drive.

**Last Update Date/Time**

Specifies the date and time when the last update occurred.

### Related Commands

Table 101 provides references to related commands.

---

*Table 101. Commands Related to QUERY DRIVE*

---

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE LIBRARY	Deletes a library.	Page 220
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
UPDATE DRIVE	Changes the attributes of a drive.	Page 575

---

## QUERY EVENT

---

### QUERY EVENT (Query Scheduled and Completed Events)

Use the QUERY EVENT command to display scheduled and completed events. The QUERY EVENT command takes two forms, depending on whether the query applies to scheduled client operations or scheduled administrative commands. The syntax and parameters for each form are defined separately.

---

### Querying Events for Client Schedules

Use the QUERY EVENT command to display scheduled and completed events for selected clients. You can specify a time range to limit output to events whose scheduled start times fall within the range.

Each scheduled client operation is called an event. The server tracks each scheduled event and records the results of each function in the database. An event record is created whenever processing of a scheduled command is started or missed.

You can request one of two different types of queries: *general* reporting or *exception* reporting. To request a general report of scheduled events, specify this command with EXCEPTIONSONLY=NO (the default). For each schedule that matches the specified policy domain and schedule name, the expected startup windows are determined within the specified time range. ADSM provides output data that indicates scheduled start times, along with any actual execution information which can be obtained from existing output records.

To request an exception report, specify EXCEPTIONSONLY=YES. ADSM provides information about scheduled operations that have *not* executed successfully, either because they have not executed as scheduled or because they have failed during execution. This mode differs from EXCEPTIONSONLY=NO, in that output is restricted as follows:

- The only events that are displayed are those that meet *one* of the following conditions:
  - The scheduled startup window has elapsed and the schedule has not begun execution.
  - The client has notified the server that the scheduled command has failed during execution and all allowed retry attempts have been exhausted.
  - The schedule began, but has not completed execution and the schedule period has elapsed since the scheduled start time.
- Information is only displayed for events that should have an output record. Thus, output records are not displayed if the records are missing but are eligible for deletion through *either* of the following:
  - Removal of output records whose retention period has elapsed. For more information on setting the event retention period, see “SET EVENTRETENTION (Specify the Retention Period for Event Records)” on page 510.

## QUERY EVENT

- Execution of a DELETE EVENT command.

No information is displayed for an event unless both of the following conditions are met:

- The last modification date for the schedule occurred before the end of the startup window for the event.
- The date on which the client was associated with the schedule occurred before the end of the startup window for the event.

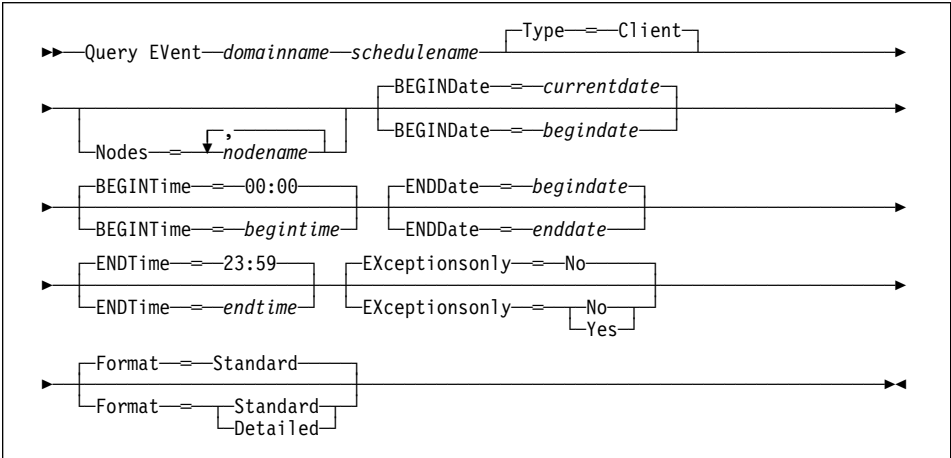
Processing of this command can require intensive computations, especially if multiple startup windows must be calculated. To minimize the processing time when querying events:

- Minimize the length of the time range for querying events.
- Restrict the policy domains, schedule names, and client node names to those for which information is required.

### Privilege Class

Any administrator can issue this command.

### Syntax for Querying Events for Client Schedules



### Parameters for Querying Events for Client Schedules

*domainname*

Specifies the name of the policy domain to which the schedules belong. This parameter is required. You can use a pattern-matching expression to specify this name. All matching policy domains are queried.

*schedulename*

Specifies the name of the schedule for which events are displayed. This parameter is required. You can use a pattern-matching expression to specify this name. All matching schedule names are queried.

## QUERY EVENT

### **Type=Client**

Specifies that this command should query events for client schedules. This parameter is optional. The default value is CLIENT. Administrative command schedule events are queried by specifying TYPE=ADMINISTRATIVE.

### **Nodes=nodename**

Specifies the name of the client node that belongs to the specified policy domain for which events are displayed. You can specify more than one client name by separating each name with a comma, with no intervening spaces. You can use a pattern-matching expression to specify a name. All matching clients are queried. If you do not specify a client name, events for all clients matching the DOMAINNAME and SCHEDULENAME parameters are displayed.

### **BEGINDate=date**

Specifies the beginning date of the time range for the events to be displayed. All events that were scheduled to start during this time are displayed. This parameter is optional. The default value is the current date.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the beginning of the time range in MM/DD/YYYY format (for example, 01/31/1995).

### **TODAY**

Specifies that the beginning of the time range is the current date.

### **TODAY-numdays or -numdays**

Specifies that the beginning of the time range is a specific number of days before the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start during the past 7 days, you can specify one of the following parameter combinations:

BEGINDATE=TODAY-7 ENDDATE=TODAY

or

BEGINDATE=-7 ENDDATE=TODAY

### **TODAY+numdays or +numdays**

Specifies that the beginning of the time range is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start two days from now, you can specify one of the following parameters:

BEGINDATE=TODAY+2

or

BEGINDATE=+2

### **BEGINTime=time**

Specifies the beginning time of the range for events to be displayed, in HH:MM format. All events scheduled to start during this time are displayed. This parameter is optional. The default value is 00:00.



## QUERY EVENT

Possible values are:

*HH:MM:SS*

Specifies the beginning of the time range on the specified begin date using HH:MM:SS format (for example, 12:33:28).

**NOW**

Specifies that the beginning of the time range is the current time on the specified begin date.

**NOW**+*numhours:numminutes* **or** +*numhours:numminutes*

Specifies that the beginning of the time range is the current time plus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start 3 hours from now, you can specify one of the following parameters:

```
BEGINTIME=NOW+03:00
or
BEGINTIME=+03:00
```

ADSM displays events at 12:00 on the specified begin date.

**NOW**-*numhours:numminutes* **or** -*numhours:numminutes*

Specifies that the beginning of the time range is the current time minus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start during the last 4 hours, you can specify one of the following parameters:

```
BEGINTIME=NOW-04:00 ENDTIME=NOW
or
BEGINTIME=-04:00 ENDTIME=NOW
```

ADSM displays events at 5:00 on the specified begin date.

**ENDDate**=*date*

Specifies the ending date of the time range for events to be displayed. All events that were scheduled to start during this time are displayed. This parameter is optional. The default value is the value used for the BEGINDATE.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the end of the time range in MM/DD/YYYY format (for example, 01/31/1995).

**TODAY**

Specifies that the end of the time range is the current date.

## QUERY EVENT

### **TODAY**-*numdays* or -*numdays*

Specifies that the end of the time range is a specific number of days before the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start during a one-week period that ended yesterday, you can specify one of the following parameter combinations:

```
BEGINDATE=TODAY-8 ENDDATE=TODAY-1
or
BEGINDATE=-8 ENDDATE=-1
```

### **TODAY**+*numdays* or +*numdays*

Specifies that the end of the time range is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start during the next week, you can specify one of the following parameter combinations:

```
BEGINDATE=TODAY ENDDATE=TODAY+7
or
BEGINDATE=TODAY ENDDATE=+7
```

### **ENDTime**=*time*

Specifies the ending time of the range for events to be displayed, in HH:MM format. All events scheduled to start during this time are displayed. This parameter is optional. The default value is 23:59.

Possible values are:

*HH:MM:SS*

Specifies the ending of the time range at a specific time on the specified end date using HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies the ending of the time range is the current time on the specified end date.

### **NOW**+*numhours:numminutes* or +*numhours:numminutes*

Specifies that the ending of the time range is the current time plus a specific number of hours (and, optionally, minutes) on the specified end date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start 3 hours from now, you can specify one of the following parameters:

```
BEGINTIME=NOW ENDTIME=NOW+03:00
or
BEGINTIME=NOW ENDTIME=+03:00
```

## QUERY EVENT

### **NOW**–*numhours:numminutes* or –*numhours:numminutes*

Specifies that the ending of the time range is the current time minus a specific number of hours (and, optionally, minutes) on the specified end date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start during the last 4 hours, you can specify one of the following parameters:

```
BEGINTIME=NOW-04:00 ENDTIME=NOW
or
BEGINTIME=-04:00 ENDTIME=NOW
```

ADSM displays events that were scheduled to start at 5:00 on the specified end date.

### **EXceptiononly**=*exceptionvalue*

Specifies the type of query information you want on scheduled or completed events. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that information on past and projected events is displayed.

#### **Yes**

Specifies that only those events which failed or did not execute as scheduled will be displayed.

### **Format**=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### **Standard**

Specifies that partial information for events is displayed.

#### **Detailed**

Specifies that complete information for events is displayed.

---

## Querying Events for Administrative Command Schedules

Use the QUERY EVENT command to display scheduled and completed events for selected administrative command schedules.

Each scheduled administrative command is called an event. The server tracks each scheduled event and records the results of each function in the database. An output record is created whenever processing of a scheduled command is started or missed.

You can request one of two different types of queries: *general* reporting or *exception* reporting. To request a general report of scheduled events, specify this command with EXCEPTIONSONLY=NO (the default). For each schedule that matches the specified schedule name, the expected startup windows are determined within the specified time

## QUERY EVENT

range. ADSM provides output data that indicates scheduled start times, along with any actual execution information which can be obtained from existing output records.

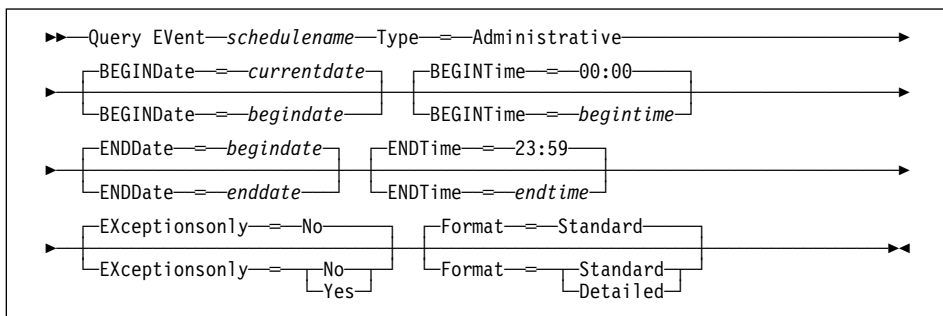
Processing of this command can require intensive computations, especially if multiple startup windows must be calculated. To minimize the processing time when querying events:

- Minimize the length of the time range for querying events.
- Restrict the schedule names to those for which information is required.

### Privilege Class

Any administrator can issue this command.

### Syntax for Querying Events for Administrative Command Schedules



### Parameters for Querying Events for Administrative Command Schedules

#### *schedulename*

Specifies the name of the schedule for which events are displayed. This parameter is required. You can use a pattern-matching expression to specify this name. All matching schedule names are queried.

#### **Type=Administrative**

Specifies that you want to query events for administrative command schedules. This parameter is required for querying administrative command schedules.

#### **BEGINDate=date**

Specifies the beginning date of the time range for the events to be displayed. All events that were scheduled to start during this time are displayed. This parameter is optional. The default value is the current date.

Possible values are:

#### *MM/DD/YYYY*

Specifies the date for the beginning of the time range in MM/DD/YYYY format (for example, 01/31/1995).

#### **TODAY**

Specifies that the beginning of the time range is the current date.

## QUERY EVENT

### **TODAY**-*numdays* or -*numdays*

Specifies that the beginning of the time range is a specific number of days before the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start during the past 7 days, you can specify one of the following parameter combinations:

```
BEGINDATE=TODAY-7 ENDDATE=TODAY
      or
BEGINDATE=-7 ENDDATE=TODAY
```

### **TODAY**+*numdays* or +*numdays*

Specifies that the beginning of the time range is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start two days from now, you can specify one of the following parameters:

```
BEGINDATE=TODAY+2
      or
BEGINDATE=+2
```

### **BEGINTIME**=*time*

Specifies the beginning time of the range for events to be displayed, in HH:MM format. All events scheduled to start during this time are displayed. This parameter is optional. The default value is 00:00.

Possible values are:

*HH:MM:SS*

Specifies the time for the beginning of the time range on the specified begin date using HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that the beginning of the time range is the current time on the specified begin date.

### **NOW**+*numhours:numminutes* or +*numhours:numminutes*

Specifies that the beginning of the time range is the current time plus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start 3 hours from now, you can specify one of the following parameters:

```
BEGINTIME=NOW+03:00
      or
BEGINTIME=+03:00
```

ADSM displays events at 12:00 on the specified begin date.

## QUERY EVENT

### **NOW**–*numhours:numminutes* or –*numhours:numminutes*

Specifies that the beginning of the time range is the current time minus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start during the last 4 hours, you can specify one of the following parameters:

```
BEGINTIME=NOW-04:00  ENDTIME=NOW
                        or
BEGINTIME=-04:00  ENDTIME=NOW
```

ADSM displays events at 5:00 on the specified begin date.

### **ENDDate**=*date*

Specifies the ending date of the time range for events to be displayed. All events that were scheduled to start during this time are displayed. This parameter is optional. The default value is the value used for the BEGINDATE parameter.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the end of the time range in MM/DD/YYYY format (for example, 01/31/1995).

### **TODAY**

Specifies that the end of the time range is the current date.

### **TODAY**–*numdays* or –*numdays*

Specifies that the end of the time range is a specific number of days before the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start during a one-week period that ended yesterday, you can specify one of the following parameter combinations:

```
BEGINDATE=TODAY-8  ENDDATE=TODAY-1
                        or
BEGINDATE=-8  ENDDATE=-1
```

### **TODAY**+*numdays* or +*numdays*

Specifies that the end of the time range is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to query events scheduled to start during the next week, you can specify one of the following parameter combinations:

```
BEGINDATE=TODAY  ENDDATE=TODAY+7
                        or
BEGINDATE=TODAY  ENDDATE=+7
```

## QUERY EVENT

### **ENDTime**=*time*

Specifies the ending time of the range for events to be displayed, in HH:MM format. All events scheduled to start during this time are displayed. This parameter is optional. The default value is 23:59.

Possible values are:

#### *HH:MM:SS*

Specifies the time for the ending of the time range on the specified end date using HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that the ending of the time range is the current time on the specified end date.

### **NOW**+*numhours:numminutes* **or** +*numhours:numminutes*

Specifies that the ending of the time range is the current time plus a specific number of hours (and, optionally, minutes) on the specified end date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start 3 hours from now, you can specify one of the following parameters:

```
BEGINTIME=NOW    ENDTIME=NOW+03:00
                or
BEGINTIME=NOW    ENDTIME=+03:00
```

### **NOW**-*numhours:numminutes* **or** -*numhours:numminutes*

Specifies that the ending of the time range is the current time minus a specific number of hours (and, optionally, minutes) on the specified end date.

For example, if you issue this command at 9:00 and you want to query events scheduled to start during the last 4 hours, you can specify one of the following parameters:

```
BEGINTIME=NOW-04:00 ENDTIME=NOW
                or
BEGINTIME=-04:00    ENDTIME=NOW
```

ADSM displays events that were scheduled to start at 5:00 on the specified end date.

### **EXceptiononly**=*exceptionvalue*

Specifies the type of query information you want on scheduled or completed events. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that information on past and projected events is displayed.

## QUERY EVENT

### Yes

Specifies that only those events which failed or did not execute as scheduled will be displayed.

### Format=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### Standard

Specifies that partial information for events is displayed.

#### Detailed

Specifies that complete information for events is displayed.

## Examples

### Task 1

Display partial information for all events scheduled for an administrative schedule named DOSADMIN. Limit the query to events that occurred on March 30, 1995 (03/30/1995).

**Command:** query eVent \* type=administrative begindate=03/30/1995  
enddate=03/30/1995

### Output:

Scheduled Start	Actual Start	Schedule Name	Status
03/30/1995 00:00:00	03/30/1995 00:00:01	DOSADMIN	Completed
03/30/1995 04:00:00	03/30/1995 04:00:01	DOSADMIN	Completed
03/30/1995 08:00:00	03/30/1995 08:00:01	DOSADMIN	Completed
03/30/1995 12:00:00		DOSADMIN	Future
03/30/1995 16:00:00		DOSADMIN	Future
03/30/1995 20:00:00		DOSADMIN	Future

See Task 2 for parameter definitions.

### Task 2

Display partial information for all events scheduled for DOMAIN1 that did not execute successfully. Limit the search to the client named JOE. Limit the events displayed to those scheduled to occur from November 11, 1995 (09/11/1995) to November 12, 1995 (09/12/1995).

**Command:** query event domain1 \* nodes=joe begindate=09/11/1995  
enddate=09/12/1995 exceptiononly=Yes



## QUERY EVENT

### Output:

Scheduled Start	Actual Start	Schedule Name	Node Name	Status
09/11/1995 01:00:00	09/11/1995 01:13:55	BACK1	JOE	Completed
09/12/1995 01:00:00		DAILYBKP	JOE	Missed

where:

#### Scheduled Start

Specifies the scheduled start date and time for the event. The scheduled start time displayed by this command does not include any randomization that can occur through the use of the SET RANDOMIZE command.

#### Actual Start

Specifies the actual date and time at which the client began execution of the scheduled operation. No information is displayed if the schedule has not started executing.

#### Schedule Name

Specifies the name of the schedule that initiated the event.

#### Node Name

Specifies the client scheduled to perform the operation.

#### Status

Specifies the completion status of the event. Possible values are:

##### Completed

Specifies that the scheduled event has been completed.

##### Missed

Specifies that the scheduled startup window for this event has passed and the schedule has not begun.

##### Failed

Specifies that the client reports a failure in executing the scheduled operation and successive retries have not succeeded.

##### Started

Specifies that the event has begun execution.

##### Restarted

Specifies that the client has retried execution of the scheduled operation.

##### Severed

Specifies that communications with the client is severed before the event can complete.

## QUERY EVENT

### Future

Specifies that the beginning of the startup window for the event is in the future.

### Pending

Specifies that the QUERY EVENT command was issued during the startup window for the event, but execution of the scheduled operation has not yet begun.

### Uncertain

Specifies that the state of the event cannot be determined. The server returns a record with Uncertain status if the QUERY EVENT command does not find an event record. An event record will not be found if the record was deleted or if the server was unavailable during the scheduled startup window (the schedule was never started). Records with Uncertain status are generated during the search; they do not reside in the database. If you do not want these records to be displayed, either specify EXCEPTIONSONLY=YES or delete the schedule if it is no longer needed.

### Task 3

Display complete information on all events that are scheduled for execution by client JOE in DOMAIN1 any time from November 11, 1995 (09/11/95) through November 12, 1995 (09/12/95).

**Command:** query event domain1 \* nodes=joe begindate=09/11/1995  
enddate=09/12/1995 format=detailed

### Output:

```
Policy Domain Name: DOMAIN1
  Schedule Name: DAILY_BACKUP
  Node Name: JOE
  Scheduled Start: 09/11/1995 01:00:00
  Actual Start:
  Completed:
  Status: Missed
  Result: 0

Policy Domain Name: DOMAIN1
  Schedule Name: DAILY_BACKUP
  Node Name: JOE
  Scheduled Start: 09/11/1995 01:00:00
  Actual Start: 09/11/1995 01:13:55
  Completed: 09/11/1995 01:18:27
  Status: Completed
  Result: 0
```

where:

### Policy Domain Name

Specifies the name of the policy domain to which the schedule is assigned.

## QUERY EVENT

**Schedule Name**

Specifies the name of the schedule that has initiated this event.

**Node Name**

Specifies the client scheduled to perform the operation.

**Scheduled Start**

Specifies the scheduled starting date and time for the event. The scheduled start time displayed does not include any randomization that can occur through the use of the SET RANDOMIZE command.

**Actual Start**

Specifies the actual date and time at which the client began execution of the scheduled operation. No information is displayed if the schedule has not started executing.

**Completed**

Specifies the date and time the scheduled event is completed.

**Status**

Specifies whether the event has been completed. Possible values are:

**Completed**

Specifies that the scheduled event has been completed.

**Missed**

Specifies that the scheduled startup window for this event has passed and the schedule has not begun.

**Failed**

Specifies that the client reports a failure in executing the scheduled operation and successive retries have not succeeded.

**Started**

Specifies that the event has begun execution.

**Restarted**

Specifies that the client has retried execution of the scheduled operation.

**Severed**

Specifies that communications with the client is severed before the event can complete.

**Future**

Specifies that the beginning of the startup window for the event is in the future.

**Pending**

Specifies that the QUERY EVENT command was issued during the startup window for the event, but execution of the scheduled operation has not yet begun.

## QUERY EVENT

### Uncertain

Specifies that the state of the event cannot be determined. The server returns a record with Uncertain status if the QUERY EVENT command does not find an event record. An event record will not be found if the record was deleted or if the server was unavailable during the scheduled startup window (the schedule was never started). Records with Uncertain status are generated during the search; they do not reside in the database. If you do not want these records to be displayed, either specify EXCEPTIONSONLY=YES or delete the schedule if it is no longer needed.

### Result

Specifies the return code from the client that identifies whether the schedule has executed successfully. If the return code is something other than a zero, examine the option string you specified using the OPTIONS parameter or the object string you specified using the OBJECTS parameter in the DEFINE SCHEDULE command.

## Related Commands

Table 102 provides references to related commands.

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DELETE EVENT	Deletes event records prior to a specified date and time.	Page 212
SET EVENTRETENTION	Specifies the number of days to retain event records.	Page 510
SET RANDOMIZE	Specifies the randomization of start times within a window for schedules in client-polling mode.	Page 526

## QUERY FILESPACE

---

### QUERY FILESPACE (Query One or More File Spaces)

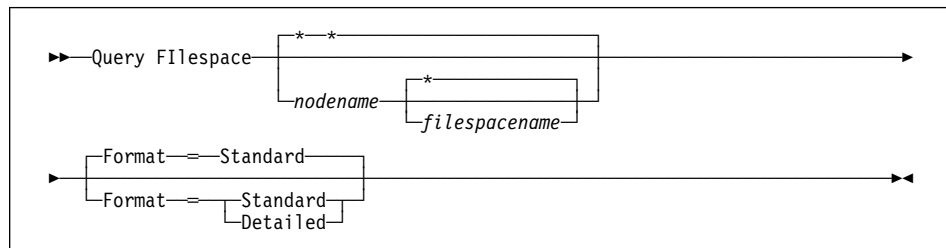
Use the QUERY FILESPACE command to display information about one or more file spaces that belong to a client node. The information displayed via the QUERY FILESPACE command reflects data as of the last incremental backup.

File space names are case sensitive. You can use the QUERY FILESPACE command to determine the correct capitalization for the file space to be queried.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *nodename*

Specifies the name of the client node to which the file space belongs. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching client node names are included. If you do not specify a value for this parameter, all client nodes are queried.

You must specify a value for this parameter if you specify FILESPACE NAME.

##### *filespace name*

Specifies the name of the file space to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching file spaces are displayed. If you do not specify a value for this parameter, all file spaces are queried.

##### **Format**=*format value*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the specified file space.

##### **Detailed**

Specifies that complete information is displayed for the specified file space.

## QUERY FILESPACE

### Examples

#### Task 1

Query all file spaces associated with all client nodes.

**Command:** query filesystem

**Output:**

Node Name	Filespace Name	Platform	Filespace Type	Capacity (MB)	%Util
JOEOS2	OS2	OS/2	HPFS	58.0	32.0
JOEOS2	D_DRIVE	OS/2	HPFS	57.0	30.8

where:

#### **Node Name**

Specifies the name of the client node.

#### **Filespace Name**

Specifies the name of the file space for the client node.

#### **Platform**

Specifies the platform for the client node.

#### **Filespace Type**

Specifies the type of file space. The type is set by the client.

#### **Capacity (MB)**

Specifies the amount of space assigned to this file space, in megabytes, on the client node. This value applies to the actual file systems and media on the client node's machine.

#### **%Util**

Specifies, as a percentage of the capacity of the file space, the amount of occupied space in the file space, on the client node. This value applies to the actual file systems and media on the client node's machine.

#### Task 2

Display detailed information on the OS2 file space that belongs to the client node JOEOS2.

**Command:** query filesystem joeos2 OS2 format=detailed

## QUERY FILESPACE

### Output:

```
Node Name: JOEOS2
Filespace Name: OS2
Platform: OS/2
Filespace Type: HPFS
Capacity (MB): 58.0
%Util: 99.7
Last Backup Start Date/Time:
Days Since Last Backup Started:
Last Backup Completion Date/Time:
Days Since Last Backup Completed:
```

where:

#### **Node Name**

Specifies the name of the client node.

#### **Filespace Name**

Specifies the name of the file space for the client node.

#### **Platform**

Specifies the platform for the client node.

#### **Filespace Type**

Specifies the type of file space. The type is set by the client.

#### **Capacity (MB)**

Specifies the amount of space assigned to this file space, in megabytes, on the client node.

#### **%Util**

Specifies, as a percentage of the capacity for the file space, the amount space that is occupied in the file space on the client node.

#### **Last Backup Start Date/Time**

Specifies the date and time that the last increment backup of the file space was initiated.

#### **Days Since Last Backup Started**

Specifies the number of days since the last increment backup of the file space was initiated.

#### **Last Backup Completion Date/Time**

Specifies the completion date and time of the last increment backup of the file space.

#### **Days Since Last Backup Completed**

Specifies the number of days since the last increment backup of the file space completed.

## QUERY FILESPACE

### Related Commands

Table 103 provides references to related commands.

*Table 103. Commands Related to QUERY FILESPACE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DELETE FILESPACE	Deletes data associated with one or more client's filespaces.	Page 216
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466
REMOVE NODE	Removes a client from the list of registered nodes for a specific policy domain.	Page 472
RENAME FILESPACE	Renames a client filesystem on the server.	Page 476
UPDATE NODE	Changes the attributes associated with a client node.	Page 585



## QUERY LIBRARY

---

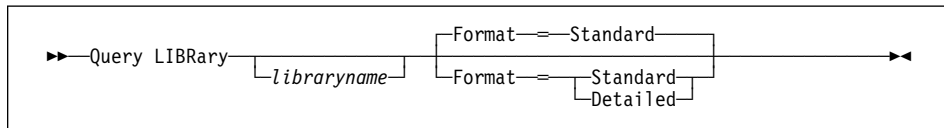
### QUERY LIBRARY (Query a Library)

Use the QUERY LIBRARY command to display information about libraries.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *libraryname*

Specifies the name of the library to be queried. You can use pattern-matching expressions to specify names. This parameter is optional.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the library.

##### **Detailed**

Specifies that complete information is displayed for the library.

#### Examples

##### Task 1

Display information about the library named AUTO.

**Command:** query library auto

**Output:**

Library Name	Library Type	Device
AUTO	SCSI	AUTO05\$

##### Task 2

Display information in full detail about the library named EZLIFE.

## QUERY LIBRARY

**Command:** query library ezlife format=detailed

**Output:**

```
Library Name: EZLIFE
Library Type: SCSI
Device: AUTO05$
External Manager:
Last Update by (administrator): DOCTOR_MIKE
Last Update Date/Time: 1995-01-13 15:24:53
```

where:

**Library Name**

Specifies the name of the library to which the drive is assigned.

**Library Type**

Specifies the type of library to which the drive is assigned.

**Device**

Specifies the name of the device that corresponds to the drive.

**Private Category**

Specifies the category number for private volumes that must be mounted by name.

**Scratch Category**

Specifies the category number to be used for scratch volumes in the library.

**Last Update by (administrator)**

Specifies who performed the last update to the drive.

**Last Update Date/Time**

Specifies the date and time when the last update occurred.

## QUERY LIBRARY

### Related Commands

Table 104 provides references to related commands.

*Table 104. Commands Related to QUERY LIBRARY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE LIBRARY	Deletes a library.	Page 220

## QUERY LIBVOLUME

---

### QUERY LIBVOLUME (Query a Library Volume)

Use the QUERY LIBVOLUME command to display information about one or more volumes that have been previously checked into an automated library for use by the ADSM server.

#### Privilege Class

Any administrator can issue this command.

#### Syntax

```
►► Query LIBVolume [libraryname] [volname] ◄◄
```

#### Parameters

##### *libraryname*

Specifies the name of the library where the storage volume is located. You can use a pattern-matching expression to specify this name. This parameter is optional.

##### *volname*

Specifies the volume name of the storage volume. You can use a pattern-matching expression to specify this name. This parameter is optional.

#### Examples

##### Task

Display information about all of the volumes that are checked into the library named AUTO.

**Command:** query libvolume auto

##### Output:

Library Name	Volume Name	Status	Last Use
AUTO	WPD001	Private	Data
AUTO	WPD002	Private	Export
AUTO	WPDV00	Scratch	
AUTO	WPD003	Private	DbBackup

##### Library Name

Specifies the name of the library where the storage volume is located.

##### Volume Name

Specifies the name of the storage volume.

## QUERY LIBVOLUME

### Status

Specifies the status (that is, private or scratch) of the storage volume.

### Last Use

Specifies the type of data on the storage volume. This field only applies to storage volumes with the Private status.

## Related Commands

Table 105 provides references to related commands.

*Table 105. Commands Related to QUERY LIBVOLUME*

Command	Description	Reference
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
CHECKIN LIBVOLUME	Checks a storage volume into an automated library.	Page 80
CHECKOUT LIBVOLUME	Checks a storage volume out of an automated library.	Page 84
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
UPDATE LIBVOLUME	Changes the status of a storage volume.	Page 579

## QUERY LICENSE

---

### QUERY LICENSE (Display License Information)

Use the QUERY LICENSE command to display information about the results of the latest license audit and the server license terms. This command also displays information indicating whether or not the server is in compliance with the terms of the license.

#### Privilege Class

Any administrator can issue this command.

#### Syntax

```
▶▶—Query LICense—————▶▶
```

#### Parameters

None.

#### Examples

##### Task

Display audit license, license terms, and compliance information.

**Command:** query license

##### Output:

```
                Last License Audit: 06/20/1996 11:52:46
                Registered Client Nodes: 34
                Licensed Client Nodes: 123
                Are network connections in use ?: No
                Are network connections licensed ?: Yes
                Are UNIX clients registered ?: No
                Are UNIX clients licensed ?: Yes
                Are desktop clients registered ?: No
                Are desktop clients licensed ?: Yes
                Are OEMVS clients registered ?: No
                Are OEMVS clients licensed ?: Yes
                Is space management in use on the server ?: No
                Is space management licensed on the server ?: Yes
                Device support module 1 required ?: Yes
                Device support module 1 licensed ?: Yes
                Device support module 2 required ?: No
                Device support module 2 licensed ?: Yes
                Device support module 3 required ?: No
                Device support module 3 licensed ?: Yes
                Device support module 4 required ?: No
                Device support module 4 licensed ?: Yes
                Server License Compliance: Valid
```

## QUERY LICENSE

where:

**Last License Audit On**

Specifies the date and time when the last license audit took place. The displayed information is valid as of this date and time.

**Registered Client Nodes**

Specifies the total number of registered client nodes.

**Licensed Client Nodes**

Specifies the total number of client nodes that can be registered as defined in the license terms.

**Are UNIX clients registered?**

Specifies whether any UNIX clients other than AIX clients are registered to the server.

**Are UNIX clients licensed?**

Specifies whether the server is licensed for UNIX clients other than AIX clients.

**Are desktop clients registered?**

Specifies whether desktop clients (such as DOS) are registered to the server.

**Are desktop clients licensed?**

Specifies whether the server is licensed to support desktop clients.

**Are Network clients registered?**

Specifies whether the network clients are registered.

**Are Network clients licensed?**

Specifies whether the network clients are licensed.

**Are OEMVS clients registered?**

Specifies whether OpenEdition MVS clients are registered with the server.

**Are OEMVS clients licensed?**

Specifies whether the server is licensed to support OpenEdition MVS clients.

**Is space management in use on the server?**

Specifies whether the server is implementing space management.

**Is space management licensed on the server?**

Specifies whether the server is licensed for space management.

**Device support module 1 required?**

Specifies whether support for Device Support Module 1 is required.

**Device support module 1 licensed?**

Specifies whether the server is licensed for Device Support Module 1.

**Device support module 2 required?**

Specifies whether support for Device Support Module 2 is required.

**Device support module 2 licensed?**

Specifies whether the server is licensed for Device Support Module 2.

## QUERY LICENSE

### **Device support module 3 required?**

Specifies whether support for Device Support Module 3 is required.

### **Device support module 3 licensed?**

Specifies whether the server is licensed for Device Support Module 3.

### **Device support module 4 required?**

Specifies whether support for Device Support Module 4 is required.

### **Device support module 4 licensed?**

Specifies whether the server is licensed for Device Support Module 4.

### **Server License Compliance**

Specifies whether the ADSM server is in compliance (Valid) or out of compliance (FAILED) with the license terms.

## Related Commands

Table 106 provides references to related commands.

*Table 106. Commands Related to QUERY LICENSE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LICENSES	Checks for compliance with defined licenses.	Page 54
QUERY AUDITOCCUPANCY	Displays the data storage utilization for a client node.	Page 325
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
REGISTER LICENSE	Registers a new license with the ADSM server.	Page 463
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.	Page 512



## QUERY LOG

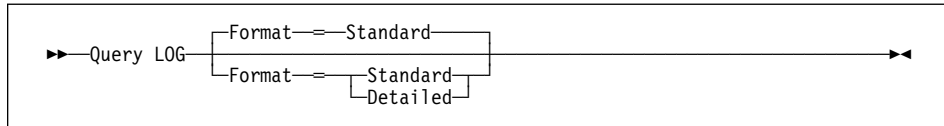
### QUERY LOG (Display Information on the Recovery Log)

Use the QUERY LOG command to display allocation information about the recovery log, including utilization, expansion, and reduction abilities.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the recovery log.

##### **Detailed**

Specifies that complete information is displayed for the recovery log.

#### Examples

##### **Task 1**

Display information about the recovery log.

**Command:** query log

**Output:**

Available Space (MB)	Assigned Capacity (MB)	Maximum Extension (MB)	Maximum Reduction (MB)	Page Size (bytes)	Total Usable Pages	Used Pages	%Util	Max. %Util
16	8	8	4	4,096	1,536	187	12.2	12.8

where:

##### **Available Space (MB)**

Specifies the total amount of space available to the recovery log, in megabytes.

## QUERY LOG

### **Assigned Capacity (MB)**

Specifies the assigned storage capacity of the recovery log, in megabytes.

### **Maximum Extension (MB)**

Specifies the maximum amount of storage space that you can extend the recovery log, in megabytes.

### **Maximum Reduction (MB)**

Specifies the maximum amount of storage space that you can reduce the recovery log, in megabytes.

### **Page Size (bytes)**

Specifies the page size, in bytes.

### **Total Usable Pages**

Specifies the total number of pages available for transactions in the recovery log. A page is a unit of allocation within recovery log volumes.

### **Used Pages**

Specifies the total number of pages currently being used by the recovery log.

### **%Util**

Specifies the current utilization as a percentage of the assigned capacity value. This percentage can vary significantly.

### **Max. %Util**

Specifies the maximum occupancy for the recovery log, as a percentage of the available space. This value specifies the highest capacity occupied by the recovery log since the last reset of this statistic. You can reset this value with the RESET LOGMAXUTILIZATION command.

### **Task 2**

Display information about the recovery log in complete detail.

**Command:** `query log format=detailed`

## QUERY LOG

### Output:

```
Available Space (MB): 16
Assigned Capacity (MB): 8
Maximum Extension (MB): 08
Maximum Reduction (MB): 4
Page Size (bytes): 4,096
Total Usable Pages: 1,536
Used Pages: 187
  %Util: 12.2
  Max. %Util: 12.8
Physical Volumes: 3
Log Pool Pages: 64
Log Pool Pct. Util: 0.31
Log Pool Pct. Wait: 0.00
Cumulative Consumption (MB): 0.60
Consumption Reset Date/Time: 05/31/1996 13:02:42
```

where:

#### **Available Space (MB)**

Specifies the amount of space available in the recovery log, in megabytes.

#### **Assigned Capacity (MB)**

Specifies the assigned storage capacity for the recovery log, in megabytes.

#### **Maximum Extension (MB)**

Specifies the maximum amount of storage space that you can extend the recovery log, in megabytes.

#### **Maximum Reduction (MB)**

Specifies the maximum amount of storage space that you can reduce the recovery log, in megabytes.

#### **Page Size (bytes)**

Specifies the page size, in bytes.

#### **Total Usable Pages**

Specifies the total number of pages available for transactions in the recovery log. A page is a unit of allocation within recovery log volumes.

#### **Used Pages**

Specifies the total number of pages currently in use by the recovery log.

#### **%Util**

Specifies the current utilization as a percentage value of the assigned capacity.

## QUERY LOG

### **Max. %Util**

Specifies the maximum occupancy for the recovery log, as a percentage of the available space. This value specifies the highest capacity occupied by the recovery log since the last reset of this statistic. You can reset this value with the RESET LOGMAXUTILIZATION command.

### **Physical Volumes**

Specifies the total number of recovery log volumes defined to the recovery log, including volume copies.

### **Log Pool Pages**

Specifies the number of pages in the recovery log buffer pool. This value is determined by the value for the LOGPOOLSIZE option. The default value for the LOGPOOLSIZE option is 128 kilobytes (32 pages).

### **Log Pool Pct. Util**

Specifies, as a percentage of all recovery log buffer pool pages, the number of recovery log buffer pool pages that are used to write changes to the recovery log after a transaction is committed.

### **Log Pool Pct. Wait**

Specifies, as a percentage of all recovery log buffer pool requests, the number of requests for a recovery log buffer pool page that is not available (because all buffer pool pages are waiting to write to the recovery log). If this number is greater than 0.0, you should increase the size of the buffer pool with the LOGPOOLSIZE option.

### **Cumulative Consumption**

Specifies the amount of log space used by the server since the last reset of this statistic. You can reset this statistic with the RESET LOGCONSUMPTION command.

### **Consumption Reset Date/Time**

Specifies the date and time of the last reset of the cumulative consumption statistic for the recovery log. You can reset this statistic with the RESET LOGCONSUMPTION command.

## QUERY LOG

### Related Commands

Table 107 provides references to related commands.

*Table 107. Commands Related to QUERY LOG*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
EXTEND DB	Extends usable database storage within the volumes previously allocated for its use.	Page 269
EXTEND LOG	Extends usable log storage within the volumes previously allocated for its use.	Page 271
REDUCE DB	Attempts to free up a database volume by reducing usable storage within the volumes defined.	Page 457
REDUCE LOG	Attempts to free up a log volume by reducing usable storage within the volumes defined.	Page 459
RESET LOGCONSUMPTION	Resets the statistic on recovery log space used.	Page 485
RESET LOGMAXUTILIZATION	Resets the maximum utilization statistic for the recovery log.	Page 487

## QUERY LOGVOLUME

---

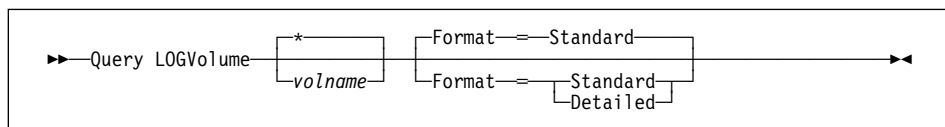
### QUERY LOGVOLUME (Display Information on One or More Log Volumes)

Use the QUERY LOGVOLUME command to display information on one or more recovery log volumes, including available, allocated, and free space on the volume. This command displays information about the specified recovery log volume and any recovery log volume copies.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *volname*

Specifies the name of the recovery log volume to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching recovery log volumes are displayed. If you do not specify a volume name, all volumes are displayed.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the specified recovery log volume.

##### **Detailed**

Specifies that complete information is displayed for the specified recovery log volume.

#### Examples

##### Task 1

Display information about the recovery log volume named log.1.

**Command:** query logvolume log.1

**Output:**

## QUERY LOGVOLUME

Volume Name (Copy 1)	Copy Status	Volume Name (Copy 2)	Copy Status	Volume Name (Copy 3)	Copy Status
c:\admserv\ log1.dsm	Sync'd		Undef- ined		Undef- ined

where:

### Volume Name (Copy *n*)

Specifies the name of the database volume. The output displays status of the three volume copies (Copy 1, Copy 2, and Copy 3).

### Copy Status

Specifies the status of the database volume.

#### Offline

The volume is varied offline.

#### Stale

The volume is not synchronized with the other volumes in the recovery log and cannot be used for storage.

#### Sync'd

The volume is synchronized with the other volumes in the recovery log and can be used for storage.

#### Undefined

No volume has been defined.

## Task 2

Display detailed information about the recovery log volume named LOG.1.

**Command:** query logvolume log.1 format=detailed

### Output:

Volume Name (Copy 1):	c:\admserv\log1.dsm
Copy Status:	Sync'd
Volume Name (Copy 2):	
Copy Status:	Undefined
Volume Name (Copy 3):	
Copy Status:	Undefined
Available Space (MB):	8
Allocated Space (MB):	8
Free Space (MB):	0

where:

## QUERY LOGVOLUME

### Volume Name (Copy *n*)

Specifies the name of the database volume. The output displays status of the three volume copies (Copy 1, Copy 2, and Copy 3).

### Copy Status

Specifies the status of the database volume. Possible values are:

#### Offline

The volume is varied offline.

#### Stale

The volume is not synchronized with the other volumes in the recovery log and cannot be used for storage.

#### Sync'd

The volume is synchronized with the other volumes in the recovery log and can be used for storage.

#### Undefined

No volume has been defined.

### Available Space (MB)

Specifies the total amount of space that is available, both allocated and free, for the recovery log, in megabytes.

### Allocated Space

Specifies the amount of space allocated for the recovery log, in megabytes.

### Free Space

Specifies the amount of unassigned space in the recovery log that can be allocated for use, in megabytes.

## Related Commands

Table 108 provides references to related commands.

Table 108. Commands Related to QUERY LOGVOLUME

Command	Description	Reference
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
DELETE LOGVOLUME	Deletes a volume from the recovery log.	Page 222
VARY	Specifies whether a disk volume is available to the server for use.	Page 629



## QUERY MGMTCLASS

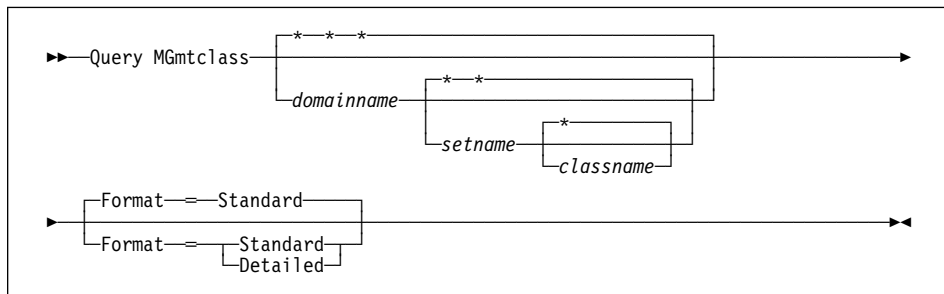
### QUERY MGMTCLASS (Query a Management Class)

Use the QUERY MGMTCLASS command to display information on an existing management class. This command is useful for determining which management classes are available, including the default management class, for a policy domain. To identify the default management class, use the QUERY DOMAIN command.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domainname*

Specifies the name of the policy domain associated with the management class to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy domains are queried. If you do not specify a value for this parameter, all policy domains are queried. You must specify this parameter when querying an explicitly named management class.

##### *setname*

Specifies the name of the policy set associated with the management class to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy sets are queried. If you do not specify a value for this parameter, all policy sets are queried. You must specify this parameter when querying an explicitly named management class.

##### *classname*

Specifies the name of the management class associated with the policy set to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching management classes are queried. If you do not specify a value for this parameter, all management classes are queried. You must specify this parameter when querying an explicitly named management class.

## QUERY MGMTCLASS

### **Format**=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### **Standard**

Specifies that partial information is displayed for the specified management class.

#### **Detailed**

Specifies that complete information is displayed for the specified management class.

## Examples

### **Task 1**

Query all management classes for all policy domains. Create the output in standard format.

**Command:** query mgmtclass

#### **Output:**

Policy Domain Name	Policy Set Name	Mgmt Class Name	Default Mgmt Class ?	Description
EMPLOYEE-RECORDS	ACTIVE	ACTIVEFILES	Yes	
EMPLOYEE-RECORDS	HOLIDAY	ACTIVEFILES	Yes	
EMPLOYEE-RECORDS	HOLIDAY	FILEHISTORY	No	
EMPLOYEE-RECORDS	VACATION	ACTIVEFILES	Yes	
EMPLOYEE-RECORDS	VACATION	FILEHISTORY	No	
PROG1	SUMMER	MCLASS1	No	Technical Support Mgmt Class
PROG2	SUMMER	MCLASS1	No	Technical Support Mgmt Class
STANDARD	ACTIVE	STANDARD	Yes	Installed default management class.
STANDARD	STANDARD	STANDARD	Yes	Installed default management class.

where:

#### **Policy Domain Name**

Specifies the name of the policy domain.

## QUERY MGMTCLASS

### Policy Set Name

Specifies the name of the activated policy set for the policy domain.

### Mgmt Class Name

Specifies the name of the management class.

### Default Mgmt Class?

Specifies whether the management class is defined as the default management class for the policy set.

### Description

Specifies a description of the management class.

### Task 2

Query the ACTIVEFILES management class that is assigned to the VACATION policy set of the EMPLOYEE\_RECORDS policy domain. Create the output in detailed format.

**Command:** query mgmtclass employee\_records vacation activefiles  
format=detailed

### Output:

```
Policy Domain Name: EMPLOYEE_RECORDS
Policy Set Name: VACATION
Mgmt Class Name: ACTIVEFILES
Default Mgmt Class?: Yes
Description:
Space Management Technique: None
Auto-Migrate on Non-Use: 0
Migration Requires Backup?: Yes
Migration Destination: SPACEMGPOOL
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 13:15:45
```

where:

### Policy Domain Name

Specifies the name of the policy domain.

### Policy Set Name

Specifies the name of the activated policy set for the policy domain.

### Mgmt Class Name?

Specifies the name of the management class.

### Default Mgmt Class

Specifies whether the management class is defined as the default management class for the policy set.

### Description

Specifies a description of the management class.

## QUERY MGMTCLASS

### Space Management Technique

Specifies the space management technique for the management class.

### Auto-Migrate on Non-Use

Specifies the number of days that must elapse since the file was last accessed before it is eligible for automatic migration.

### Migration Requires Backup?

Specifies whether a backup version of a file must exist before a file can be migrated.

### Migration Destination

Specifies the name of an ADSM storage pool that is a destination for migrated files.

### Last Update by (administrator)

Specifies the name of the administrator that defined or most recently updated the management class.

### Last Update Date/Time

Specifies the date and time that the administrator defined or most recently updated the management class.

## Related Commands

Table 109 provides references to related commands.

*Table 109. Commands Related to QUERY MGMTCLASS*

Command	Description	Reference
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
UPDATE MGMTCLASS	Changes the attributes of a management class.	Page 581

## QUERY MOUNT

---

### QUERY MOUNT (Display Information on Mounted Sequential Access Volumes)

Use the QUERY MOUNT command from the administrative client to request information about the status of one or all sequential access volumes that are mounted.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

*volname*

Specifies the name of the mounted sequential access volume that you want to query. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching mounted volumes are queried. If you do not specify a sequential access volume name, information on all mounted volumes is displayed.

#### Examples

##### Task

Display information on the mounted sequential media volume named WPD000.

**Command:** query mount wpd000

##### Output:

```
ANR8331I 8MMTAPE volume WPD000 is mounted R/W  
in drive 8MM.1 (tape01$), status: DISMOUNTING.  
ANR8334I 1 volume(s) found.
```

## QUERY MOUNT

### Related Commands

Table 110 provides references to related commands.

*Table 110. Commands Related to QUERY MOUNT*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DISMOUNT VOLUME	Dismounts a sequential, removable volume by the volume name.	Page 245
REPLY	Allows a request to continue processing.	Page 481

## QUERY NODE

---

### QUERY NODE (Query One or More Client Nodes)

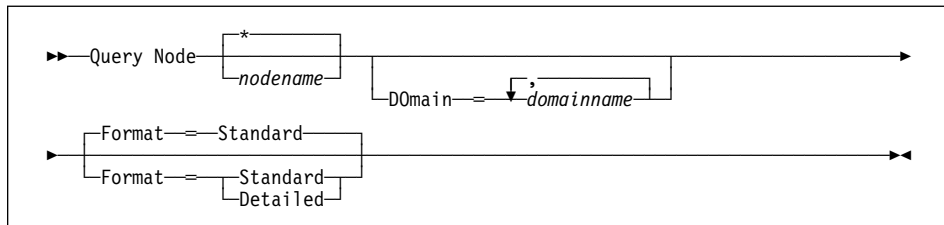
Use the QUERY NODE command to display information about one or more registered client nodes. This command provides client node information such as:

- The name of the client node
- The date the server was last accessed
- The date the password was last changed
- The client node's policy domain
- The communication method used by the client node

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

#### *nodename*

Specifies the name of the client node to be queried. You can use a pattern-matching expression to specify this name. All matching client nodes are queried. If you do not specify a value for this parameter, all client nodes are queried.

#### **D0main=***domainname*

Specifies a list of policy domains that limit the client node query. Only nodes that are assigned to one of the specified policy domains are displayed. This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use a pattern-matching expression to specify a domain. All clients assigned to a matching domain will be displayed. If you do not specify a value for this parameter, all policy domains are included in the query.

#### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the specified client nodes.

##### **Detailed**

Specifies that complete information is displayed for the specified client nodes.

## QUERY NODE

### Examples

#### Task 1

Display information on all registered client nodes.

**Command:** query node

**Output:**

Node Name	Platform	Policy Domain Name	Days Since Last Access	Days Since Password Set	Locked?
CLIENT1	(?)	STANDARD	6	6	No
GEORGE	OS/2	STANDARD	1	1	No
JANET	(?)	STANDARD	1	1	No
JOEOS2	OS/2	STANDARD	<1	<1	No
TOMC	(?)	STANDARD	1	1	No

where:

#### Node Name

Specifies the name of the client node.

#### Platform

Specifies the platform of the client node. A question mark (?) is displayed until the client node first accesses the server and reports its platform type.

**Note:** After a node first accesses the server and reports its platform type, the platform type will be constant, even if the node platform subsequently changes.

#### Policy Domain Name

Specifies the name of the policy domain to which the client node is assigned.

#### Days Since Last Access

Specifies the number of days that have elapsed since the last time the client node accessed the server.

#### Days Since Password Set

Specifies the number of days that have elapsed since the client node's password was set.

#### Locked?

Specifies whether the client node is locked out of ADSM.



## QUERY NODE

### Task 2

Display complete information on the client node named JOEOS2.

**Command:** query node joeos2 format=detailed

**Output:**

```
Node Name: JOEOS2
Platform: OS/2
Policy Domain Name: STANDARD
Last Access Date/Time: 05/19/1995 09:59:52
Days Since Last Access: 2
Password Set Date/Time: 05/18/1995 11:04:45
Days Since Password Set: 3
Locked?: No
Contact:
Compression: No
Archive Delete Allowed?: Yes
Backup Delete Allowed?: No
Registration Date/Time: 05/18/1995 11:04:45
Registering Administrator: OPEN REGISTRATION
Last Communication Method Used: Tcp/Ip
Bytes Received Last Session: 226
Bytes Sent Last Session: 556
Duration of Last Session (sec): 3.32
Pct. Idle Wait Last Session: 88.48
Pct. Comm. Wait Last Session: 6.63
Pct. Media Wait Last Session: 0.00
```

where:

**Node Name**

Specifies the name of the client node.

**Platform**

Specifies the platform of the client node. A question mark (?) is displayed until the client node first accesses the server and reports its platform type.

**Policy Domain Name**

Specifies the name of the policy domain to which the client node is assigned.

**Last Access Date/Time**

Specifies the last date and time that the client node accessed the server.

**Days Since Last Access**

Specifies the number of days that have elapsed since the last time the client node accessed the server.

**Password Set Date/Time**

Specifies the date and time that the client node's password was set.

## QUERY NODE

### **Days Since Password Set**

Specifies the number of days that have elapsed since the client node's password was set.

### **Locked?**

Specifies whether the client node is locked out of ADSM.

### **Contact**

Specifies any contact information for the client node.

### **Compression**

Specifies whether compression is enabled.

### **Archive Delete Allowed?**

Specifies whether the client node can delete his or her own archive files.

### **Backup Delete Allowed?**

Specifies whether the client node can delete his or her own backup files.

### **Registration Date/Time**

Specifies the date and time that the client node was registered.

### **Registering Administrator**

Specifies the name of the administrator that registered the client node.

### **Last Communication Method Used**

Specifies the communication method that was last used by the client node.

### **Bytes Received Last Session**

Specifies the number of bytes received by the server during the last client node session.

### **Bytes Sent Last Session**

Specifies the number of bytes sent to the client node.

### **Duration of Last Session (sec)**

Specifies how long the most recent client node session lasted.

### **Pct. Idle Wait Last Session**

Specifies, as a percentage of the total session time, the amount of time that the client was not performing any functions.

### **Pct. Comm. Wait Last Session**

Specifies, as a percentage of the total session time, the amount of time that the client waited for a communication response from the client.

### **Pct. Media Wait Last Session**

Specifies, as a percentage of the total session time, the amount of time that the client waited for a removable volume to be mounted.

## QUERY NODE

### Related Commands

Table 111 provides references to related commands.

*Table 111. Commands Related to QUERY NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
LOCK NODE	Prevents a client from accessing the server.	Page 304
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466
REMOVE NODE	Removes a client from the list of registered nodes for a specific policy domain.	Page 472
RENAME NODE	Changes the name for an existing user that belongs to a specific policy domain.	Page 479
UNLOCK NODE	Enables a locked user in a specific policy domain to access the server.	Page 536
UPDATE NODE	Changes the attributes associated with a client node.	Page 585

## QUERY OCCUPANCY

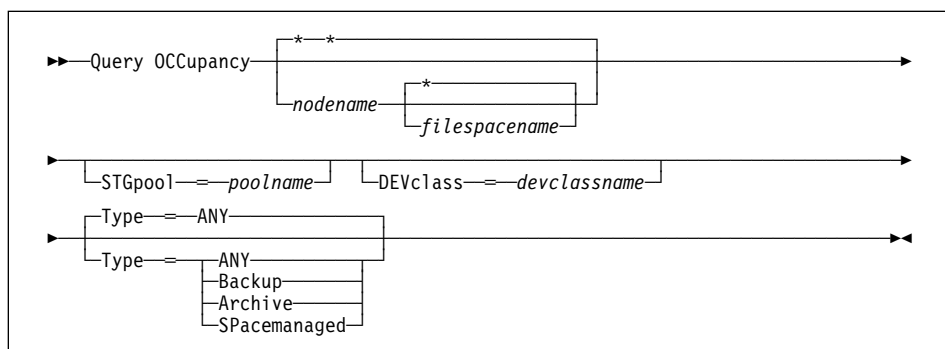
### QUERY OCCUPANCY (Display File Space Information by Storage Pool)

Use the QUERY OCCUPANCY command to display information on where a client's file spaces are stored and how much space they occupy. This command can be used to display information about file spaces that are stored in primary or copy storage pools.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *nodename*

Specifies the name of the node that owns the file spaces that you want to locate. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching names are included in the query. If you do not specify a value for this parameter, all nodes are queried.

##### *file spacename*

Specifies the name of the file space that you want to locate. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching names are included in the query. If you do not specify a value for this parameter, all file spaces are queried. If you explicitly specify a file space name, you must also specify a node name.

##### **STGpool=poolname**

Specifies the name of the storage pool in which files from the specified file space are stored. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching storage pools are included in the query. If you do not specify a value for this parameter, all storage pools are queried.

##### **DEVclass=devclassname**

Specifies the name of the device class to which the file spaces are assigned. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching device classes are included in the query. If you do not specify a value for this parameter, all storage pools are queried.

## QUERY OCCUPANCY

**Type=***filetype*

Specifies the types of files to query in the file spaces. This parameter is optional. The default value is ANY. Possible values are:

**ANY**

Specifies that all types of files are queried: backup versions of files, archived copies of files, and files migrated from one or more client nodes.

**Backup**

Specifies that backup files are queried.

**Archive**

Specifies that archive files are queried.

**SPacemanaged**

Specifies that space management files (files migrated from one or more client nodes) are queried.

### Examples

**Task 1**

Display information on where all file spaces assigned to the node named BILLK are stored.

**Command:** query occupancy billk

**Output:**

Node Name	Filespace Name	Storage Pool Name	Number of Files	Space Occupied (MB)
BILLK	IBMDOS_5	POOL2	21	1.02
BILLK	WP_FILES	POOLA	33	7.60
BILLK	SP_DATA	BACKUPPOOL	2	0.80

where:

**Node Name**

Specifies the name of the node that owns the specified file space.

**Filespace Name**

Specifies the name of the file space that belongs to the node.

**Storage Pool Name**

Specifies the name of the storage pool where the files are stored.

**Number of Files**

Specifies the number of files that belong to the file space and are stored in this storage pool.

## QUERY OCCUPANCY

### Space Occupied (MB)

Specifies the amount of space occupied by the file space.

### Task 2

Display information on where the file spaces that belong to the node WAYNE are stored.

**Command:** query occupancy wayne

### Output:

Node Name	Filespace Name	Storage Pool Name	Number of Files	Space Occupied (MB)
WAYNE	DWG1	BACKUPPOOL1	2,330	53.19
WAYNE	OS2C	BACKUPPOOL1	1,554	32.00
WAYNE	OS2D	BACKUPPOOL1	1,716	35.30

where:

### Node Name

Specifies the name of the node that owns the specified file space.

### Filespace Name

Specifies the name of the file space that belongs to the node.

### Storage Pool Name

Specifies the name of the storage pool where files are stored.

### Number of Files

Specifies the number of files that belong to the file space and are stored in this storage pool.

### Space Occupied (MB)

Specifies the amount of space occupied by the file space.

## Related Commands

Table 112 provides references to related commands.

Table 112 (Page 1 of 2). Commands Related to QUERY OCCUPANCY

Command	Description	Reference
DELETE FILESPACE	Deletes data associated with one or more client's filespaces.	Page 216

## QUERY OCCUPANCY

*Table 112 (Page 2 of 2). Commands Related to QUERY OCCUPANCY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY FILESPACE	Displays information about data in one or more tablespaces that belong to a client.	Page 377
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403

## QUERY OPTION

---

### QUERY OPTION (Query One or More Server Options)

Use the QUERY OPTION command to display information about one or more general server options.

The options file is the only place in which the server options can be changed. To change the options, you must restart the server.

For more information about the server options, see Appendix E, “Tailoring Server Options” on page 677.

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

*optionname*

Specifies the name of an option in the server options file for which you want information. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching server options are displayed. If you do not specify this parameter, information on all options is displayed.

### Examples

#### Task 1

Display general information about all defined server options.

**Command:** query option



## QUERY OPTION

### Output:

Server Option	Option Setting	Server Option	Option Setting
CommTimeOut	60	IdleTimeOut	15
BufPoolSize	2048	LogPoolSize	512
DateFormat	1 (mm/dd/yyyy)	TimeFormat	1 (hh:mm:ss)
NumberFormat	1 (1,000.00)	MessageFormat	1
Language	AMENG	MaxSessions	25
ExpInterval	24	ExpQuiet	No
MirrorRead DB	Normal	MirrorRead LOG	Normal
MirrorWrite DB	Sequential	MirrorWrite LOG	Parallel
VolumeHistory	beta	VolumeHistory	gama
Devconfig	lan	Devconfig	hoa
TxnGroupMax	40	MoveBatchSize	40
MoveSizeThresh	500	StatusMsgCnt	10
TcpPort	1574	TCPWindowSize	0
TCPNoDelay	Yes	IPXSocket	1574
NetBiosName	NETBLM16	NetbiosBufferSize	32768
NetbiosSessions	6	LuName	DSMSERV1
SHMQUEUEENAME	\QUEUES\ADSM\DSMSERV	SnaBufferSize	15360
IpxBufferSize	32768	NamedPipeName	\PIPE\DSMSERV
LanAdapter	0	CommMethod	TCPIP
CommMethod	SNAlu6.2	CommMethod	NETBIOS
CommMethod	NAMEDPIPE	CommMethod	TLISPX
CommMethod	SHAREDMEMORY	Message Interval	1
MaxMemory	536870912		

where:

#### Server Option

Specifies the name of the server option.

#### Option Setting

Specifies the current value defined to the server option.

## QUERY OPTION

### Task 2

View the option settings for all options that begin with L.

**Command:** query option l\*

**Output:**

Server Option	Option Setting
LogPoolSize	256
Language	AMENG
LuName	DSMSERV1
LanAdapter	2

where:

**Server Option**

Specifies the name of the server option.

**Option Setting**

Specifies the current value defined to the server option.

### Related Commands

None.

## QUERY POLICYSET

---

### QUERY POLICYSET (Query a Policy Set)

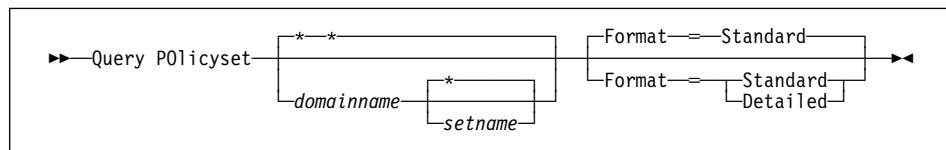
Use the QUERY POLICYSET command to display information about an existing policy set.

To get a list of the policy sets for a policy domain, use the QUERY DOMAIN command.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domainname*

Specifies the name of the policy domain associated with the policy set to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy domains are queried. If you do not specify a value for this parameter, all policy domains are queried. You must specify this parameter when querying an explicitly named policy set.

##### *setname*

Specifies the name of the policy set associated with the copy group to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching policy sets are queried. If you do not specify either ACTIVE or the policy set name, all policy sets are queried. You must specify this parameter when querying an explicitly named policy set.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the policy set.

##### **Detailed**

Specifies that complete information is displayed for the policy set.

#### Examples

##### Task 1

Query all policy sets for all policy domains. Create the output in standard format.

**Command:** query policyset

## QUERY POLICYSET

### Output:

Policy Domain Name	Policy Set Name	Default Mgmt Class Name	Description
EMPLOYEE- _RECORDS	ACTIVE	ACTIVEFI- LES	
EMPLOYEE- _RECORDS	HOLIDAY	ACTIVEFI- LES	
EMPLOYEE- _RECORDS	VACATION	ACTIVEFI- LES	
PROG1	SUMMER		Programming Group Policies
PROG2	SUMMER		Programming Group Policies
STANDARD	ACTIVE	STANDARD	Installed default policy set.
STANDARD	STANDARD	STANDARD	Installed default policy set.

where:

#### Policy Domain Name

Specifies the name of the policy domain.

#### Policy Set Name

Specifies the name of the policy set.

#### Default Mgmt Class Name

Specifies the assigned default management class for the policy set.

#### Description

Specifies a description of the policy domain.

### Task 2

Query the VACATION policy set that is defined to the EMPLOYEE\_RECORDS policy domain. Create the output in detailed format.

**Command:** `query policyset employee_records vacation format=detailed`

### Output:

## QUERY POLICYSET

```
Policy Domain Name: EMPLOYEE_RECORDS
Policy Set Name: VACATION
Default Mgmt Class Name: ACTIVEFILES
Description:
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 13:15:50
```

where:

**Policy Domain Name**

Specifies the name of the policy domain.

**Policy Set Name**

Specifies the name of the activated policy set for the policy domain.

**Default Mgmt Class Name**

Specifies the assigned default management class for the policy set.

**Description**

Specifies a description of the policy domain.

**Last Update by (administrator)**

Specifies the name of the administrator that defined or most recently updated the policy domain.

**Last Update Date/Time**

Specifies when the administrator defined or most recently updated the storage pool.

## QUERY POLICYSET

### Related Commands

Table 113 provides references to related commands.

*Table 113. Commands Related to QUERY POLICYSET*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY POLICYSET	Creates a copy of an existing policy set.	Page 94
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354
UPDATE POLICYSET	Changes the description of a policy set.	Page 588
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## QUERY PROCESS

---

### QUERY PROCESS (Query One or More Server Processes)

Use the QUERY PROCESS command to display information about one or more active background processes. The following commands can generate background processes:

AUDIT LIBRARY	EXPORT NODE
AUDIT LICENSES	EXPORT POLICY
AUDIT VOLUME	EXPORT SERVER
BACKUP DB	EXTEND LOG
BACKUP STGPOOL	IMPORT ADMIN
CHECKIN LIBVOLUME	IMPORT NODE
CHECKOUT LIBVOLUME	IMPORT POLICY
DEFINE DBCOPY	IMPORT SERVER
DEFINE LOGCOPY	MOVE DATA
DELETE DBVOLUME	REDUCE DB
DELETE FILESPACE	REDUCE LOG
DELETE LOGVOLUME	RESTORE STGPOOL
DELETE VOLUME	RESTORE VOLUME
EXPIRE INVENTORY	VARY
EXPORT ADMIN	

The following internal server operations generate background server processes:

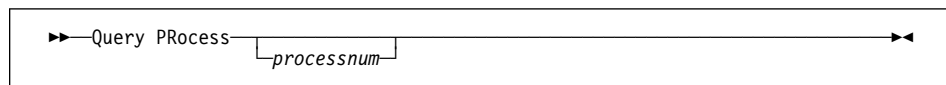
- Inventory expiration
- Migration
- Reclamation

You can cancel background processes, by process number, using the CANCEL PROCESS command.

### Privilege Class

Any administrator can issue this command.

### Syntax



### Parameters

*processnum*

Specifies the number of the background process to be queried. This parameter is optional. If you do not specify a process number, information on all background processes is displayed.

## QUERY PROCESS

### Examples

#### Task

Display information about background process 202 that was initiated with the EXPORT SERVER command.

**Command:** query process 202

#### Output:

Process Number	Process Description	Status
202	EXPORT SERVER	ANR0648I Have copied the following: 8 Domains 2 Policy Sets 10 Management Classes 4 Copy Groups 1 Administrators 746 Bytes (0 errors have been detected)

where:

#### Process Number

Specifies the number assigned to the background process.

#### Process Description

Specifies a description of the background process.

#### Status

Specifies the completion status of the background process.

### Related Commands

Table 114 provides a reference to a related command.

Table 114. Command Related to QUERY PROCESS

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74



## QUERY REQUEST

---

### QUERY REQUEST (Query One or More Pending Mount Requests)

Use the QUERY REQUEST command to display information about one or more pending mount requests.

#### Privilege Class

Any administrator can issue this command.

#### Syntax

```
► Query REQuest requestnum ◄◄
```

#### Parameters

*requestnum*

Specifies the number of the pending mount request. This parameter is optional. If you do not specify a value for this parameter, all pending mount requests are displayed.

#### Examples

##### Task 1

Display information on all pending mount requests.

**Command:** query request

**Output:**

```
ANR8352I Requests outstanding:
ANR8326I 001: Mount 8MM volume EXP001 R/W
in drive 8MM.1 (tape01$) of library
MANLIB within 60 minute(s).
```

#### Related Commands

Table 115 provides references to related commands.

Table 115. Related Commands for QUERY REQUEST

Command	Description	Reference
CANCEL REQUEST	Cancels one or more pending volume mount requests.	Page 76
REPLY	Allows a request to continue processing.	Page 481

## QUERY SCHEDULE

---

### QUERY SCHEDULE (Query One or More Schedules)

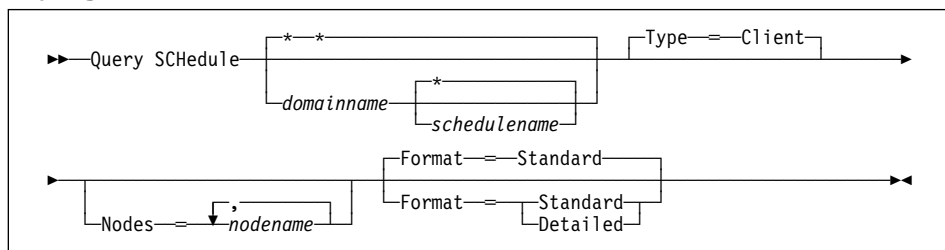
Use the QUERY SCHEDULE command to display information pertaining to one or more schedules.

The QUERY SCHEDULE command takes two forms, depending on whether the schedule applies to client operations or administrative commands. The syntax and parameters for each form are defined separately.

#### Privilege Class

Any administrator can issue this command.

#### Syntax for Querying Client Schedules



#### Parameters for Querying Client Schedules

##### *domainname*

Specifies the name of the policy domain to which the schedule belongs. You can use a pattern-matching expression to specify this name. All matching policy domains are queried. The default queries all existing policy domains. If you specify a domain name, you do not have to specify a schedule name.

##### *schedulingname*

Specifies the name of the schedule that belongs to the specified policy domain. You can use a pattern-matching expression to specify this name. All matching schedules are queried. The default value queries all existing schedules for a given domain. If you specify a schedule name, you must also specify a policy domain name.

##### **Type=Client**

Specifies that this command should query client schedules. This parameter is optional. The default value is CLIENT. Administrative command schedules are queried by specifying TYPE=ADMINISTRATIVE.

##### **Nodes=nodename**

Specifies that you want to restrict the schedules that are queried by defining one or more client nodes that are associated with the schedules to be displayed. This parameter is optional. You can use a pattern-matching expression to specify a name. All matching clients are queried. If you do not specify a client name, all schedules matching the DOMAINNAME and SCHEDULENAME

## QUERY SCHEDULE

parameters are displayed. You can specify more than one client node name by separating each name with a comma, with no intervening spaces.

**Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

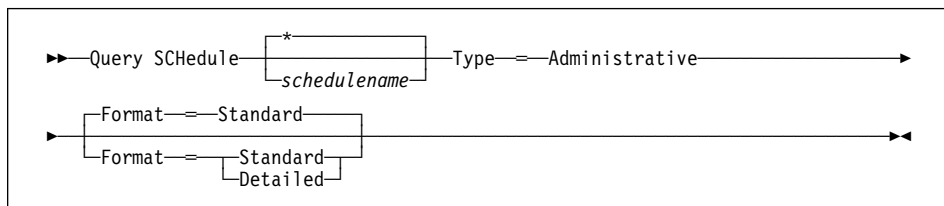
**Standard**

Specifies that partial information is displayed for the specified schedules.

**Detailed**

Specifies that complete information is displayed for the specified schedules.

### Syntax for Querying Administrative Command Schedules



### Parameters for Querying Administrative Command Schedules

*schemulename*

Specifies the name of the schedules to be queried. You can use a pattern-matching expression to specify this name. All matching schedules are queried. The default queries all existing schedules.

**Type=Administrative**

Specifies that the schedules to be queried are administrative schedules. This parameter is required.

**Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

**Standard**

Specifies that partial information is displayed for the specified schedules.

**Detailed**

Specifies that complete information is displayed for the specified schedules.

### Examples

**Task 1**

Display all schedules that belong to the EMPLOYEE\_RECORDS policy domain.

**Command:** query schedule employee\_records

## QUERY SCHEDULE

### Output:

Domain	Schedule Name	Action	Start Date/Time	Duration	Period	Day
EMPLOYEE_RE- CORDS	WEEKLY_BACKUP	Inc Bk	07/04/1994 23:00:00	4 H	1 W	Sat

where:

#### Domain

Specifies the name of the policy domain to which the specified schedule belongs.

#### \* (Asterisk)

Specifies whether the corresponding schedule has expired. If there is an asterisk in this column, the corresponding schedule has expired.

#### Schedule Name

Specifies the name of the schedule.

#### Action

Specifies the action that occurs when this schedule executes.

#### Start Date/Time

Specifies the initial starting date and time for this schedule.

#### Duration

Specifies the length of the startup window.

#### Period

Specifies the time between startup windows (assuming DAYOFWEEK=ANY).

#### Day

Specifies the day of the week on which the startup windows begin.

### Task 2

Display, in detail, the administrative command schedule (BACKUP\_BACKUPPOOL) defined to backup the primary storage pool BACKUPPOOL to the copy storage pool COPYSTG.

**Command:** query schedule backup\_backuppool type=administrative  
format=detailed

## QUERY SCHEDULE

### Output:

```
Schedule Name: BACKUP_BACKUPPOOL
Description:
  Command: backup stgpool backuppool copystg
  Priority: 5
Start Date/Time: 05/31/1996 20:00:00
  Duration: 1 Hour(s)
  Period: 2 Day(s)
  Day of Week: Any
Expiration:
  Active?: Yes
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/1996 14:03:30
```

where:

#### **Schedule Name**

Specifies the name of the schedule.

#### **Description**

Specifies the description of the schedule.

#### **Options**

Specifies the options supplied to the DSMC command at the time the schedule is executed.

#### **Command**

Specifies the command that is scheduled.

#### **Priority**

Specifies the priority value for this schedule.

#### **Start Date/Time**

Specifies the initial starting date and time for this schedule.

#### **Duration**

Specifies the length of the startup window.

#### **Period**

Specifies the time between startup windows (assuming DAYOFWEEK=ANY).

#### **Day of Week**

Specifies the day of the week on which the startup windows begin.

#### **Expiration**

Specifies the date and time on which this schedule will no longer be used. If this column is blank, the schedule does not expire.

#### **Last Update by (administrator)**

Specifies the name of the administrator that most recently modified the schedule.

## QUERY SCHEDULE

### Last Update Date/Time

Specifies the last date and time the schedule was modified.

### Active?

Specifies whether the schedule has been processed according to the time and date set for this schedule.

## Related Commands

Table 116 provides references to related commands.

*Table 116. Command Related to QUERY SCHEDULE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
COPY SCHEDULE	Creates a copy of an existing schedule.	Page 96
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164
UPDATE SCHEDULE	Changes the attributes of a schedule.	Page 590

## QUERY SESSION

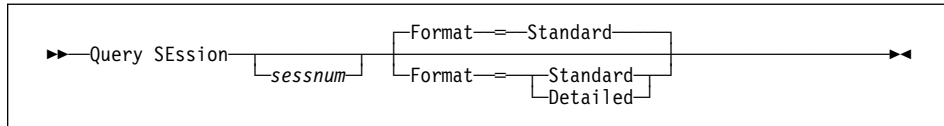
### QUERY SESSION (Query One or More Client Sessions)

Use the QUERY SESSION command to display information on one or more administrative and client node sessions.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *sessnum*

Specifies the number of the administrative or client node session to be queried. This parameter is optional. If you do not specify a value for this parameter, all sessions are displayed.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the session.

##### **Detailed**

Specifies that complete information is displayed for the session.

#### Examples

##### Task 1

Display information on all administrative and client node sessions that are communicating with the server.

**Command:** query session

**Output:**

Sess Number	Comm. Method	Sess State	Wait Time	Bytes Sent	Bytes Recvd	Sess Type	Platform	Client Name
77	Tcp/Ip	Run	0 S	134	159	Admin	OS/2	ADMIN

where:

## QUERY SESSION

### **Sess Number**

Specifies a unique session identification number assigned by the server.

### **Comm. Method**

Specifies the method used by the client to communicate with the server.

### **Sess State**

Specifies the current communications state of the server. Possible states are:

<b>End</b>	The session is ending (session resources are released).
<b>IdleW</b>	Waiting for client's next request (session is <i>idle</i> ).
<b>MediaW</b>	The session is waiting for access to a sequential access volume.
<b>RecvW</b>	The server is waiting to receive an expected message from the client.
<b>Run</b>	The server is executing a client request (and not waiting to send data).
<b>SendW</b>	The server is waiting to send data to the client (waiting for data to be delivered to the client node that has already been sent).
<b>Start</b>	The session is starting (authentication is in progress).

### **Wait Time**

Specifies the amount of time (seconds, minutes, or hours) the server has been in the current state shown.

### **Bytes Sent**

Specifies the number of bytes of data sent to the client node since the session was initiated.

### **Bytes Recvd**

Specifies the number of bytes of data received from the client node since the session was initiated.

### **Sess Type**

Specifies the type of session in process: either ADMIN for an administrative session, or NODE for a client node session.

### **Platform**

Specifies the type of operating system associated with the client.

### **Client Name**

Specifies the name of the administrator or client node.

### **Task 2**

Display information in full detail about all administrative and client node sessions that are communicating with the server.



## QUERY SESSION

**Command:** Query SSession Format=Detailed

**Output:**

```
Sess Number: 77
Comm. Method: Tcp/Ip
Sess State: Run
Wait Time: 0 S
Bytes Sent: 409
Bytes Recvd: 202
Sess Type: Admin
Platform: OS/2
Client Name: ADMIN
Media Access Status:
User Name:
```

where:

**Sess Number**

Specifies a unique session identification number assigned by the server.

**Comm. Method**

Specifies the method being used by the client to communicate with the server.

**Sess State**

Specifies the current communications state of the server. Possible states are:

- |               |  |
|---------------|--|
| <b>End</b>    | The session is ending (session resources are released).  |
| <b>IdleW</b>  | Waiting for client's next request (session is <i>idle</i> ).   |
| <b>MediaW</b> | The session is waiting for access to a sequential access volume.   |
| <b>RecvW</b>  | Waiting to receive an expected message from the client.  |
| <b>Run</b>    | The server is executing a client request (and not waiting to send data).   |
| <b>SendW</b>  | The server is waiting to send data to the client (waiting for data to be delivered to the client node that has already been sent). |
| <b>Start</b>  | The session is starting (authentication is in progress).   |

**Wait Time**

Specifies the amount of time (seconds, minutes, or hours) the server has been in the current state shown.

**Bytes Sent**

Specifies the number of bytes of data sent to the client node since the session was initiated.

## QUERY SESSION

**Bytes Recvd**

Specifies the number of bytes of data received from the client node since the session was initiated.

**Sess Type**

Specifies the type of session in process: either ADMIN for an administrative session, or NODE for a client node session.

**Platform**

Specifies the type of operating system associated with the client.

**Client Name**

Specifies the name of the administrator or client node.

**Media Access Status**

Specifies the type of media wait state. This information is only provided when the session state equals *MediaW*.

### Related Commands

Table 117 provides a reference to a related command.

*Table 117. Command Related to QUERY SESSION*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL SESSION	Cancels one or all active sessions with the server.	Page 78

## QUERY STATUS

---

### QUERY STATUS (Query System Parameters)

Use the QUERY STATUS command to display information about the general server parameters, such as those defined by the SET commands.

You can use the QUERY STATUS command to request information about client sessions, such as the availability of the server, password authentication, accounting settings, or the retention period for the information retained in the activity log. You can also use this command to display information about the central scheduler, such as the central scheduling mode of the server, or to display the maximum number of retries allowed after a failed attempt to execute a scheduled command.

### Privilege Class

Any administrator can issue this command.

### Syntax

```
▶—Query Status—◀
```

### Parameters

None.

### Examples

#### Task

Display general information about server parameters.

**Command:** query status

**Output:**

## QUERY STATUS

```
Server Name: WELLS_DESIGN_DEPT
Server Installation Date/Time: 05/31/1996 13:02:05
Server Restart Date/Time: 05/31/1996 13:14:55
Authentication: On
Password Expiration Period: 45 Day(s)
Registration: Open
Availability: Enabled
Accounting: On
Activity Log Retention Period: 30 Day(s)
License Audit Period: 14 Day(s)
Last License Audit: 05/31/1996 13:02:55
Server License Compliance: Valid
Central Scheduler: Active
Maximum Sessions: 16
Maximum Scheduled Sessions: 4
Event Record Retention Period: 15 Day(s)
Schedule Randomization Percentage: 50
Query Schedule Period: Client's Choice
Maximum Command Retries: Client's Choice
Retry Period: 15 Minute(s)
Scheduling Modes: Any
Log Mode: RollForward
Database Backup Trigger: Not Defined
```

where:

### **Server Name**

Specifies the name of the server.

### **Server Installation Date/Time**

Specifies the date and time the server was installed.

### **Server Restart Date/Time**

Specifies the last date and time the server was started.

### **Authentication**

Specifies whether password authentication is set on or off.

### **Password Expiration Period**

Specifies the period, in days, after which the administrator or client node password expires.

### **Registration**

Specifies whether client node registration is open or closed.

### **Availability**

Specifies whether the server is enabled or disabled.

### **Accounting**

Specifies whether an accounting record is generated at the end of each client node session.

## QUERY STATUS

### **Activity Log Retention Period**

Specifies the number of days information is retained in the activity log.

### **License Audit Period**

Specifies the period, in days, after which the license manager automatically audits the ADSM license. Additional licensing information is available by using the QUERY LICENSE command.

### **Last License Audit**

Specifies the date and time when the last license audit occurred. Additional licensing information is available by using the QUERY LICENSE command.

### **Server License Compliance**

Specifies whether the server is in compliance (Valid) or out of compliance (FAILED) with the license terms.

### **Central Scheduler**

Specifies whether central scheduling is running (active or aborted).

### **Maximum Sessions**

Specifies the maximum number of client/server sessions.

### **Maximum Scheduled Sessions**

Specifies the maximum number of client/server sessions available for processing scheduled work.

### **Event Record Retention Period**

Specifies the number of days central scheduler event records are retained.

### **Schedule Randomization Percentage**

Specifies how much of the startup window is used for executing scheduled events in client-polling mode.

### **Query Schedule Period**

Specifies the frequency with which clients poll the server to obtain scheduled work, in client-polling mode. If the value in this field is *Client's Choice*, the polling frequency is determined by the client node.

### **Maximum Command Retries**

Specifies the maximum number of times that a client scheduler retries after a failed attempt to execute a scheduled command. If the value in this field is *Client's Choice*, the client node determines the maximum number of retries.

### **Retry Period**

Specifies the number of minutes between attempts by the client scheduler to retry after a failed attempt to contact the server or to execute a scheduled command. If the value in this field is *Client's Choice*, the client node determines the number of minutes between retries.

### **Scheduling Modes**

Specifies the central scheduling modes supported by the server.

## QUERY STATUS

### Log Mode

Specifies the mode for saving recovery log records. The mode is either roll-forward or normal.

### Database Backup Trigger

Specifies that ADSM automatically runs a backup of the database if the database backup trigger is enabled.

## Related Commands

Table 118 provides references to related commands.

*Table 118 (Page 1 of 2). Commands Related to QUERY STATUS*

Command	Description	Reference
DISABLE	Prevents new clients from accessing ADSM but permits existing sessions to continue.	Page 243
ENABLE	Resumes server activity following the DISABLE command.	Page 246
QUERY LICENSE	Displays information about licenses and audits.	Page 386
SET ACCOUNTING	Specifies whether accounting records are created at the end of each client session.	Page 505
SET ACTLOGRETENTION	Specifies the number of days to retain log records in the activity log.	Page 507
SET AUTHENTICATION	Specifies whether clients are prompted for a password at the beginning of a session.	Page 509
SET EVENTRETENTION	Specifies the number of days to retain event records.	Page 510
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514
SET MAXCMDRETRIES	Specifies the maximum number of retries after a failed attempt to execute a scheduled command.	Page 517

## QUERY STATUS

Table 118 (Page 2 of 2). Commands Related to QUERY STATUS

Command	Description	Reference
SET MAXSCHEDESESSIONS	Specifies the maximum number of client/server sessions available for processing scheduled work.	Page 519
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.	Page 521
SET QUERYSCHEDPERIOD	Specifies the frequency for clients to obtain scheduled work, in client-polling mode.	Page 524
SET RANDOMIZE	Specifies the randomization of start times within a window for schedules in client-polling mode.	Page 526
SET REGISTRATION	Specifies whether users can register themselves or must be registered by an administrator.	Page 528
SET RETRYPERIOD	Specifies the time between retry attempts by the client scheduler.	Page 530
SET SCHEDMODES	Specifies the central scheduling mode for the server.	Page 532
SET SERVERNAME	Specifies the name by which the server is identified.	Page 534

## QUERY STGPOOL

---

### QUERY STGPOOL (Query One or More Storage Pools)

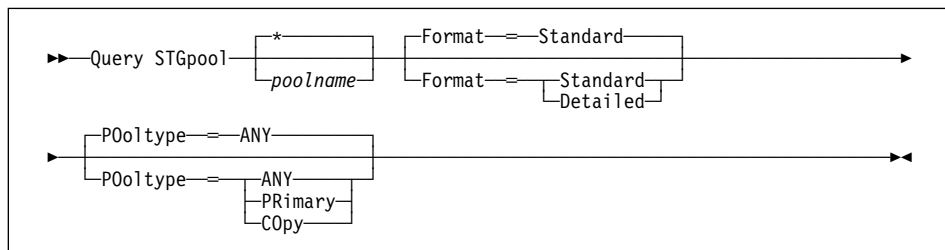
Use the QUERY STGPOOL command to display information about one or more storage pools.

You can also use this command to monitor migration processes for the storage pool.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *poolname*

Specifies the name of the storage pool to be queried. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching storage pools are included. If you do not specify a value for this parameter, all storage pools are displayed.

##### **Format=***formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the storage pool.

##### **Detailed**

Specifies that complete information is displayed for the storage pool.

##### **POoltype=***type*

Specifies the type of storage pool to be queried. This parameter is optional. The default value is ANY. Possible values are:

##### **ANY**

Specifies that primary and copy storage pools will be queried.

##### **PRimary**

Specifies that only primary storage pools will be queried.

##### **COpy**

Specifies that only copy storage pools will be queried.



## QUERY STGPOOL

### Usage Notes

1. For storage pools defined to DISK devices, if CACHE=YES, the value for %UTIL is consistently high, even after migration, as the device's capacity is maximized with cached copies of data. The value for %MIGR drops after migration, indicating that less data is available to be migrated.
2. For storage pools defined to DISK devices, the value for %UTIL can be higher than the value for %MIGR if you issue this command while a file creation transaction is in progress. The value for %UTIL is determined by the amount of space actually allocated (while the transaction is in progress) and the value for %MIGR only represents the space occupied by *committed* files. At the end of the transaction, these values become synchronized.

### Examples

#### Task 1

Display information for the storage pool named POOL1.

**Command:** query stgpool pool1

**Output:**

Storage Pool Name	Device Class Name	Estimated Capacity (MB)	%Util	%Migr	High Mig%	Low Mig%	Next Storage Pool
POOL1	DISK	58.5	0.8	0.7	90	70	POOL2

where:

#### Storage Pool Name

Specifies the name of the storage pool.

#### Device Class Name

Specifies the name of the device class assigned to the storage pool.

#### Estimated Capacity (MB)

Specifies the estimated capacity of the storage pool, in megabytes.

For DISK devices, this is the capacity of all volumes in the storage pool, including any volumes that are varied offline.

For sequential access devices, this is an estimate of the total space of all the sequential access volumes in the storage pool, including any volumes that are *unavailable* or *readonly* and all volumes defined with the MAXSCRATCH parameter.

#### %Util

Specifies, as a percentage, an estimate of the utilization of the storage pool.

For DISK devices, this is specified as a percentage of the value for the estimated

## QUERY STGPOOL

capacity, including cached data and data that resides on any volumes which are varied offline.

For sequential access devices, this is specified as a percentage of the number of active bytes on each sequential access volume and the estimated capacity of all volumes in the storage pool.

The %UTIL value includes cached data on a volume (when cache is enabled); the %MIGR value excludes cached data. Therefore, when cache is enabled and migration occurs, the %MIGR value decreases but the %UTIL value remains the same because the migrated data is remaining on the volume as cached data. In this instance, the %UTIL value decreases only when the cached data expires or when the space that cached files are using needs to be used for noncached files.

### **%Migr**

Specifies an estimate of the percentage of data in the storage pool that can be migrated. This value is used by the high and low migration thresholds to determine when to start and stop migration.

For DISK devices, this value is specified as a percentage of the value for the estimated capacity, excluding cached data but including data on any volumes varied offline.

For sequential access devices, this value is specified as the percentage of the number of sequential access volumes in the storage pool that contain at least 1 byte of active data.

The %UTIL value includes cached data on a volume (when cache is enabled); the %MIGR value excludes cached data. Therefore, when cache is enabled and migration occurs, the %MIGR value decreases but the %UTIL value remains the same because the migrated data is remaining on the volume as cached data. In this instance, the %UTIL value decreases only when the cached data expires or when the space that cached files are using needs to be used for noncached files.

### **High Mig%**

Specifies when ADSM can begin migration for the storage pool. Migration processes can begin when the percentage of data that can be migrated reaches this threshold.

### **Low Mig%**

Specifies when ADSM can stop migration for the storage pool. Migration processes can stop when the percentage of data that can be migrated reaches this threshold.

### **Next Storage Pool**

Specifies the destination storage pool for data that is migrated from this storage pool.

### **Task 2**

Display information in full detail for the storage pool named 8MMPOOL.

**Command:** `query stgpool 8mmpool format=detailed`

## QUERY STGPOOL

### Output:

```
Storage Pool Name: 8MMPool
Storage Pool Type: Primary
Device Class Name: 8MMTAPE
Estimated Capacity (MB): 0.0
    %Util: 0.0
    %Migr: 0.0
    High Mig%: 90
    Low Mig%: 70
Migration Processes:
    Next Storage Pool: POOL1
Maximum Size Threshold: 5 M
    Access: Read/Write
    Description: Copy Storage Pool
Cache Migrated Files?:
    Collocate?: Yes
Reclamation Threshold: 60
Maximum Scratch Volumes Allowed: 5
    Delay Period for Volume Reuse: 0 Day(s)
    Migration in Progress?: No
    Amount Migrated (MB): 0.00
Elapsed Migration Time (seconds): 0
    Reclamation in Progress?: No
Volume Being Migrated/Reclaimed:
    Last Update by (administrator): ADMIN
    Last Update Date/Time: 05/22/1996 14:50:29
```

where:

#### **Storage Pool Name**

Specifies the name of the storage pool.

#### **Storage Pool Type**

Specifies the type of storage pool.

#### **Device Class Name**

Specifies the name of the device class assigned to the storage pool.

#### **Estimated Capacity (MB)**

Specifies the estimated capacity of the storage pool, in megabytes.

For DISK devices, this is the capacity of all volumes in the storage pool, including any volumes that are varied offline.

For sequential access devices, this is an estimate of the total space of all the sequential access volumes in the storage pool, including volumes that are *unavailable* or *readonly* and all volumes defined with the MAXSCRATCH parameter.

## QUERY STGPOOL

### **%Util**

Specifies, as a percentage, an estimate of the utilization of the storage pool.

For DISK devices, this is specified as a percentage of the value for the estimated capacity, including cached data and data that resides on any volumes which are varied offline.

For sequential access devices, this is specified as a percentage of the number of active bytes on each sequential access volume and the estimated capacity of all volumes in the storage pool.

The %UTIL value includes cached data on a volume (when cache is enabled); the %MIGR value excludes cached data. Therefore, when cache is enabled and migration occurs, the %MIGR value decreases but the %UTIL value remains the same because the migrated data is remaining on the volume as cached data. In this instance, the %UTIL value decreases only when the cached data expires or when the space that cached files are using needs to be used for noncached files.

### **%Migr**

Specifies an estimate of the percentage of data in the storage pool that can be migrated. This value is used by the high and low migration thresholds to determine when to start and stop migration.

For DISK devices, this value is specified as a percentage of the value for the estimated capacity, excluding cached data, but including data on any volumes varied offline.

For sequential access devices, this value is specified as the percentage of the number of sequential access volumes in the storage pool that contain at least 1 byte of active data.

The %UTIL value includes cached data on a volume (when cache is enabled); the %MIGR value excludes cached data. Therefore, when cache is enabled and migration occurs, the %MIGR value decreases but the %UTIL value remains the same because the migrated data is remaining on the volume as cached data. In this instance, the %UTIL value decreases only when the cached data expires or when the space that cached files are using needs to be used for noncached files.

### **High Mig%**

Specifies when ADSM can begin migration for the storage pool. Migration processes can begin when the percentage of data that can be migrated reaches this threshold.

### **Low Mig%**

Specifies when ADSM can stop migration for the storage pool. Migration processes can stop when the percentage of data on the volume that can be migrated reaches this threshold.

### **Migration Process**

For random access storage pools only, specifies the number of processes that are used for migrating files from a storage pool. Each migration process uses a drive so the number of migration processes should not exceed the number of drives that can be dedicated to migration processing.

## QUERY STGPOOL

### **Next Storage Pool**

Specifies the destination storage pool for data that is migrated from this storage pool.

### **Maximum Size Threshold**

Specifies the maximum file size for files in the storage pool.

### **Access**

Specifies how the data in the storage pool can be accessed.

### **Description**

Specifies a description of the storage pool.

### **Cache Migrated Files?**

For DISK device classes only, specifies whether caching is enabled.

### **Collocate?**

For sequential access device classes, specifies whether collocation is enabled.

### **Reclamation Threshold**

Specifies when data from volumes in a storage pool is reclaimed. Each individual volume's percentage of reclaimable space is compared to this value to determine if reclamation is necessary.

### **Maximum Scratch Volumes Allowed**

Specifies the maximum number of scratch volumes that can be requested by ADSM.

### **Delay Period for Volume Reuse**

Specifies the number of days that must elapse after files have been deleted from a volume, before the volume can be returned to scratch or reused.

### **Migration in Progress?**

Specifies whether at least one migration process is active for the storage pool.

### **Amount Migrated (MB)**

Specifies the amount of data, in megabytes, that has been migrated, if migration is in progress. If migration is not in progress, this value indicates the amount of data migrated during the last migration. When multiple, parallel migration processes are used for the storage pool, this value indicates the total amount of data migrated by all processes.

### **Elapsed Migration Time (seconds)**

Specifies the amount of time that has elapsed since migration began, if migration is active. If migration is not active, this value indicates the amount of time required to complete the last migration. When multiple, parallel migration processes are used for the storage pool, this value indicates the total time from the beginning of the first process until the completion of the last process.

### **Reclamation in Progress?**

Specifies whether a reclamation process is active for the storage pool.

## QUERY STGPOOL

### Volume Being Migrated/Reclaimed

For sequential access storage pools, specifies the name of a sequential access volume for which migration or reclamation processing is active. If no migration or reclamation processes are active on a volume, this field is blank. For random access storage pools, this field is always blank.

### Last Update by (administrator)

Specifies the name of the administrator that has defined or most recently updated the storage pool.

### Last Update Date/Time

Specifies when the administrator has defined or most recently updated the storage pool.

## Related Commands

Table 119 provides references to related commands.

*Table 119. Commands Related to QUERY STGPOOL*

Command	Description	Reference
DEFINE STGPOOL	Defines a storage pool as a named collection of data storage media.	Page 180
DELETE STGPOOL	Deletes a storage pool from data storage.	Page 232
UPDATE STGPOOL	Changes the attributes of a storage pool.	Page 605

## QUERY VOLHISTORY

### QUERY VOLHISTORY (Display Sequential Volume History Information)

Use the QUERY VOLHISTORY command to display sequential volume history information that has been collected by the server.

Volume history information includes data such as date and time of use for the following types of volumes:

- Volumes not in storage pools
  - Volumes used to back up an ADSM database (full or incremental)
  - Volumes used to dump an ADSM database
  - Volumes used to export administrator, node, policy, or server data
- Volumes in storage pools
  - Storage pool sequential access volumes that have been added
  - Storage pool sequential access volumes that have been reused in storage pools through reclamation or MOVE DATA operations
  - Storage pool sequential access volumes that have been removed by using the DELETE VOLUME command or removed during reclamation of scratch volumes

Sequential access volumes are volumes that are accessed sequentially by the ADSM server, such as removable tape or optical volumes.

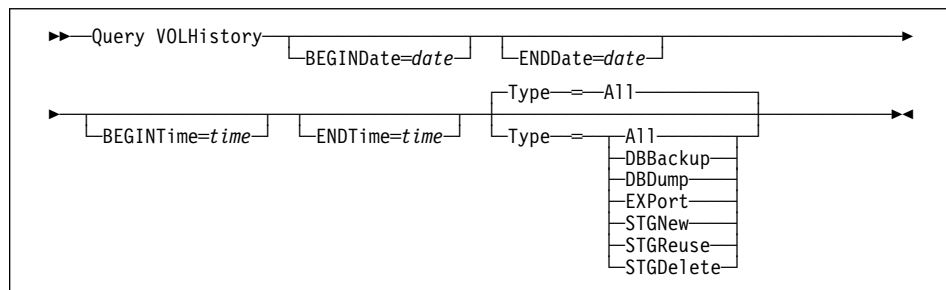
To save sequential volume history information collected by the server to one or more files, use the BACKUP VOLHISTORY command.

To delete sequential volume history information collected by the server, use the DELETE VOLHISTORY command.

### Privilege Class

Any administrator can issue this command.

### Syntax



## QUERY VOLHISTORY

### **BEGINDate=***date*

Specifies the beginning date for the history information to display. This parameter is optional. The default is the earliest date for which history information exists. Possible values are:

*MM/DD/YYYY*

Specifies that you want to display information beginning with records created on a specific date, using MM/DD/YYYY format (for example, 01/31/1995).

### **TODAY**

Specifies that you want to display information beginning with records created on the current date.

### **TODAY-***numdays* **or** *-numdays*

Specifies that you want to display information beginning with records created on the current date minus a specific number of days. The maximum value for the number of days is 9999.

For example, to display information beginning with records created a week ago, you can specify TODAY-7 or simply -7.

### **ENDDate=***date*

Specifies the ending date for the history information to display. This parameter is optional. The default is the current date. Possible values are:

*MM/DD/YYYY*

Specifies that you want to display information ending with records created on a specific date, using MM/DD/YYYY format (for example 02/28/95).

### **TODAY**

Specifies that you want to display information ending with records created on the current date.

### **TODAY-***numdays* **or** *-numdays*

Specifies that you want to display information ending with records created on or before the current date minus a specific number of days. The maximum value for the number of days is 9999.

For example, to display all volume history information created up to yesterday, you can specify TODAY-1 or simply -1.

### **BEGINTime=***time*

Specifies the beginning time for the history information to display. This parameter is optional. The default is midnight (00:00:00) on the date specified with the BEGINDATE parameter. Possible values are:

*HH:MM:SS*

Specifies that you want to display information beginning with records created at a specific time on the specified begin date using HH:MM:SS format (for example, 12:33:28).



## QUERY VOLHISTORY

### **NOW**

Specifies that you want to display information beginning with records created on the current time on the specified begin date.

### **NOW+numhours:numminutes or +numhours:numminutes**

Specifies that you want to display information beginning with records created at the current time plus a specific number of hours (and, optionally, minutes) on the specified begin date.

For example, if you issue the QUERY VOLHISTORY command at 9:00 with `BEGINTIME=NOW+3` or `BEGINTIME=+3`, ADSM displays records with a time of 12:00 or later on the specified begin date.

### **NOW-numhours:numminutes or -numhours:numminutes**

Specifies that you want to display information beginning with records created at the current time minus a specific number of hours (and, optionally minutes) on the specified begin date.

For example, if you issue the QUERY VOLHISTORY command at 9:00 with `BEGINTIME=NOW-3:30` or `BEGINTIME=-3:30`, ADSM displays records with a time of 5:30 or later on the specified begin date.

### **ENDTime=time**

Specifies the ending time for the history information to display. This parameter is optional. The default is the current time.

### **HH:MM:SS**

Specifies that you want to display information ending with records created at a specific time on the specified end date using HH:MM:SS format (for example, 18:59:59).

### **NOW**

Specifies that you want to display information ending with records created at the current time on the specified end date.

### **NOW+numhours:numminutes or +numhours:numminutes**

Specifies that you want to display information ending with records created a specific number of hours (and, optionally, minutes) after the current time on the specified end date.

For example, if you issue the QUERY VOLHISTORY command at 9:00 with `ENDTIME=NOW+3` or `ENDTIME=+3`, ADSM displays records with a time of 12:00 or earlier on the specified end date.

The following are sample commands to perform that task:

```
query volhistory enddate=now-1 endtime=now+3
query volhistory enddate=-1 endtime=+3
```

### **NOW-numhours:numminutes or -numhours:numminutes**

Specifies that you want to display information ending with records created a specific number of hours (and, optionally, minutes) prior to the current time on the specified end date.

## QUERY VOLHISTORY

For example, if you issue the QUERY VOLHISTORY command at 9:00 with ENDTIME=NOW-3:30 or ENDTIME=-3:30, ADSM displays records with a time of 5:30 or earlier on the specified end date.

### **Type=***typevalue*

Specifies the type of records to display from the volume history file. This parameter is optional. The default is ALL. Possible values are:

#### **All**

Specifies that all history information is displayed.

#### **DBBackup**

Specifies to display only records that contain information about volumes used for backups (full or incremental).

#### **DBDump**

Specifies to display only records that contain information about volumes used for online database dumps (done before installation of Version 2 Release 1).

#### **EXPort**

Specifies to display only records that contain information about volumes used for exporting data.

#### **STGNew**

Specifies to display only records that contain information about new sequential access storage volumes.

#### **STGReuse**

Specifies to display only records that contain information about reused sequential storage pool volumes.

#### **STGDelete**

Specifies to display only records that contain information about deleted sequential storage pool volumes.

## Examples

### **Task**

Display volume history information stored in the ADSM database.

**Command:** `query volhistory`

## QUERY VOLHISTORY

### Output:

```
      Date/Time: 04/06/1995 18:28:06
      Volume Type: STGNEW
      Backup Series: 0
Backup Operation: 0
      Volume Seq: 0
      Device Class: FILE
      Volume Name: GLM3A
      Volume Location:

      Date/Time: 04/06/1995 18:28:17
      Volume Type: STGNEW
      Backup Series: 0
Backup Operation: 0
      Volume Seq: 0
      Device Class: FILE
      Volume Name: GLM4A
      Volume Location:

      Date/Time: 04/06/1995 18:50:53
      Volume Type: STGNEW
      Backup Series: 0
Backup Operation: 0
      Volume Seq: 0
      Device Class: FILE
      Volume Name: GLM5A
      Volume Location:
```

where:

#### **Date/Time**

The date and time for the volume information.

#### **Volume Type**

The type of volume. Possible values are:

##### **BACKUPFULL**

The volume was used for a full backup of the database.

##### **BACKUPINCR**

The volume was used for an incremental backup of the database.

##### **DBDUMP**

The volume was used for an online dump of the database. (These volumes were used before the installation of ADSM Version 2 Release 1.)

##### **EXPORT**

The volume was used to record data for export.

##### **STGDELETE**

The volume was a sequential access storage pool volume that was deleted.

## QUERY VOLHISTORY

### **STGNEW**

The volume was a sequential access storage pool volume that was added.

### **STGREUSE**

The volume was a sequential access storage pool volume that was reused.

### **Backup Series**

If the volume type is BACKUPFULL or BACKUPINCR, the identifying number of the backup series of which this volume is a part. Each backup series consists of a full backup and all incremental backups that apply to that full backup, up to the next full backup of the database.

### **Backup Operation**

If the volume type is BACKUPFULL or BACKUPINCR, the operation number of this backup volume, within the backup series. The full backup within a backup series is operation 0. The first incremental backup for that full backup is operation 1, the second incremental backup is operation 2, and so on.

### **Volume Seq**

If the volume type is BACKUPFULL, BACKUPINCR, or EXPORT, the sequence number of the volume when it was used for backing up the database or for exporting data.

For BACKUPFULL and BACKUPINCR volumes, this is the sequence, or position, of the volume within the backup series. Volume sequence 1 identifies the first volume used for the first operation (a full backup), and so on. For example, if the full backup occupies three volumes, these volumes are identified as volume sequence 1, 2, and 3, respectively. The first volume of the next operation (the first incremental backup) is then volume sequence 4.

### **Device Class**

The device class used for the volume.

### **Volume Name**

The name of the volume.

### **Volume Location**

The location of the volume. The information is available only for BACKUPFULL, BACKUPINCR, DBDUMP, and EXPORT volumes, and only if the UPDATE VOLHISTORY command was used to add the information.

## QUERY VOLHISTORY

### Related Commands

Table 120 provides references to related commands.

*Table 120. Commands Related to QUERY VOLHISTORY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP VOLHISTORY	Records sequential volume history information in external files.	Page 72
DELETE VOLHISTORY	Removes sequential volume history information from the volume history file.	Page 234
UPDATE VOLHISTORY	Adds or changes location information for a volume in the volume history file.	Page 617

## QUERY VOLUME

---

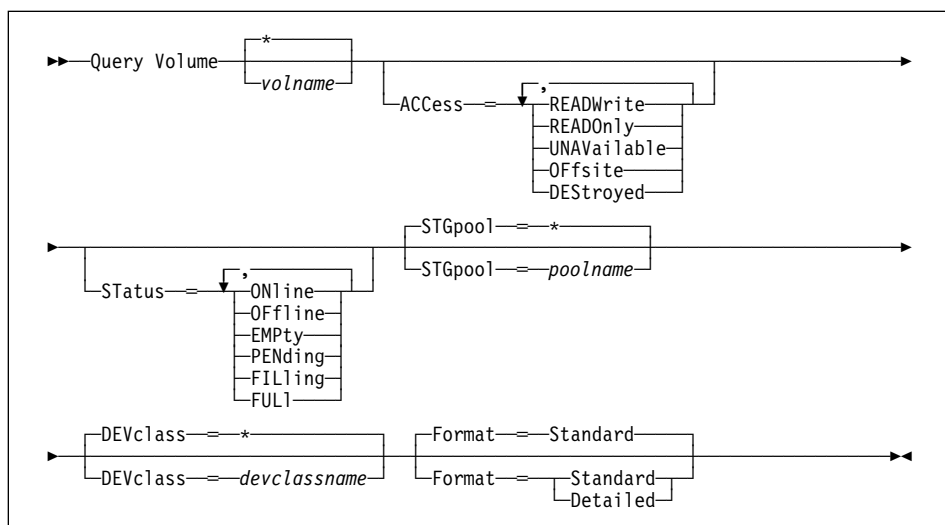
### QUERY VOLUME (Query One or More Storage Pool Volumes)

Use the QUERY VOLUME command to display information about one or more storage pool volumes. This command displays information about scratch volumes defined to ADSM.

#### Privilege Class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *volname*

Specifies the name of the volume to be displayed. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching volumes are included in the query. If you do not specify a storage pool volume name, all storage pool volumes are included in the query.

##### **ACCess=***accessmode*

Specifies that output should be restricted by volume access mode. This parameter is optional. You can specify more than one access mode in a list by delimiting each mode with a comma and no intervening spaces. If you do not specify a value for this parameter, output is not restricted by access mode. Possible values are:

##### **READWrite**

Specifies that volumes with an access mode of READWRITE are displayed.

## QUERY VOLUME

**READOnly**

Specifies that volumes with an access mode of READONLY are displayed.

**UNAVailable**

Specifies that volumes with an access mode of UNAVAILABLE are displayed.

**OFFsite**

Specifies that volumes with an access mode of OFFSITE are displayed.

**DESTroyed**

Specifies that volumes with an access mode of DESTROYED are displayed.

**Status=*status***

Specifies that output should be restricted by volume status. This parameter is optional. You can specify more than one status in a list by delimiting each status with a comma and no intervening spaces. If you do not specify a value for this parameter, output is not restricted by volume status. Possible values are:

**ONline**

Specifies that volumes with a status of ONLINE are displayed.

**OFFline**

Specifies that volumes with a status of OFFLINE are displayed.

**EMPTy**

Specifies that volumes with a status of EMPTY are displayed.

**PENDING**

Specifies that volumes with a status of PENDING are displayed. These are volumes from which all files have been deleted, but the time specified by the REUSEDELAY parameter from the DEFINE STGPOOL command has not elapsed.

**FILLing**

Specifies that volumes with a status of FILLING are displayed.

**FULI**

Specifies that volumes with a status of FULL are displayed.

**STGPool=*poolname***

Specifies the name of the storage pool to be included in the query. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching storage pools are included in the query. If you do not specify a storage pool name, all storage pools are included in the query.

**DEVclass=*devclassname***

Specifies the name of the device class to be included in the query. This parameter is optional. You can use a pattern-matching expression to specify this name. All matching device classes are included in the query. If you do not specify a device class name, all devices are included in the query.

## QUERY VOLUME

### **Format**=*formatvalue*

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### **Standard**

Specifies that partial information is displayed for the storage pool volume.

#### **Detailed**

Specifies that complete information is displayed for the storage pool volume.

## Examples

### **Task 1**

Display information on all storage pool volumes prefixed with the name ATF.

**Command:** query volume atf\*

#### **Output:**

Volume Name	Storage Pool Name	Device Class Name	Estimated Capacity (MB)	%Util	Volume Status
ATF001	8MMPool	8MMTAPE	4944.0	18.2	Filling
ATF002	8MMPool	8MMTAPE	4944.0	18.2	Filling

where:

#### **Volume Name**

Specifies the name of the storage pool volume.

#### **Storage Pool Name**

Specifies the name of the storage pool to which the volume is defined.

#### **Device Class Name**

Specifies the name of the device class assigned to the storage pool.

#### **Estimated Capacity (MB)**

Specifies the estimated capacity of the volume, in megabytes.

For DISK devices, this value is the capacity of the volume.

For sequential access devices, this value is an estimate of the total space of the volume.

#### **%Util**

Specifies an estimate of the utilization of the volume.

For DISK devices, this value is specified as a percentage of the value for the estimated capacity, including cached data.

For sequential access devices, this value is specified as a percentage of the number of active bytes on the volume and the estimated capacity of the volume.



## QUERY VOLUME

### Volume Status

Specifies whether the storage pool volume is available to ADSM.

### Task 2

Display information in full detail about the storage pool volume name WPDV00.

**Command:** query volume wpdv00 format=detailed

### Output:

```
Volume Name: WPDV00
Storage Pool Name: TAPEPOOL
Device Class Name: TAPE
Estimated Capacity (MB): 4944.0
    %Util: 18.2
Volume Status: On-line
Access: Read/Write
Pct. Reclaimable Space: 3.2
Scratch Volume?: Yes
In Error State?: No
Number of Writable Sides: 1
Number of Times Mounted: 11
Write Pass Number: 1
Approx. Date Last Written: 04/14/1996 16:17:26
Approx. Date Last Read: 04/01/1996 13:26:18
Date Became Pending:
Number of Write Errors: 0
Number of Read Errors: 0
Volume Location:
Last Update by (administrator): COLLIN
Last Update Date/Time: 05/01/1996 14:07:27
```

where:

### Volume Name

Specifies the name of the storage pool volume.

### Storage Pool Name

Specifies the name of the storage pool to which the volume is defined.

### Device Class Name

Specifies the name of the device class assigned to the storage pool.

### Estimated Capacity (MB)

Specifies the estimated capacity of the volume, in megabytes.

For DISK devices, this value is the capacity of the volume.

For sequential access devices, this value is an estimate of the total space of the volume.

## QUERY VOLUME

### **%Util**

Specifies an estimate of the utilization of the volume.

For DISK devices, this value is specified as a percentage of the value for the estimated capacity, including cached data.

For sequential access devices, this value is specified as a percentage of the number of active bytes on the volume and the estimated capacity of the volume.

### **Volume Status**

Specifies whether the volume is available to ADSM.

### **Access**

Specifies whether the volume is available to ADSM.

### **Pct. Reclaimable Space**

Specifies the amount of space on this volume that can be reclaimed. This value is compared to the value for the RECLAIM parameter (in DEFINE STGPOOL) to determine if reclamation is necessary.

### **Scratch Volume?**

Specifies whether the volume is a scratch volume.

### **In Error State?**

Specifies whether the volume is in an error state. Volumes in an error state cannot be written to.

### **Number of Writable Sides**

This information is reserved for ADSM.

### **Number of Times Mounted**

Specifies the number of times the volume has been mounted.

### **Write Pass Number**

Specifies the number of times the volume has been written to from the beginning to the end of the sequential access medium.

### **Approx. Date Last Written**

Specifies the approximate date on which the volume was last written.

### **Approx. Date Last Read**

Specifies the approximate date on which the volume was last read.

### **Date Became Pending**

Specifies the date that the volume's status changed to pending.

### **Number of Write Errors**

Specifies the number of writing errors that have occurred on the volume.

### **Number of Read Errors**

Specifies the number of reading errors that have occurred on the volume.

### **Volume Location**

Specifies the location of the volume.

## QUERY VOLUME

**Last Update by (administrator)**

Specifies the name of the administrator that has defined or most recently updated the volume.

**Last Update Date/Time**

Specifies when the volume was defined or most recently updated.

### Related Commands

Table 121 provides references to related commands.

*Table 121. Commands Related to QUERY VOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
UPDATE VOLUME	Updates the attributes of one or more storage pool volumes.	Page 619
VARY	Specifies whether a disk volume is available to the server for use.	Page 629

## QUIT

---

### QUIT (End the Interactive Mode of the Administrative Client)

Use the QUIT command to end an administrative client session in interactive mode.

You cannot use the QUIT command from the SERVER\_CONSOLE administrative ID, or the console, or the batch, or the mount modes of the administrative client.

For more information on the administrative client, see “Using the Administrative Client Session” on page 13.

### Privilege Class

Any administrator can issue this command.

### Syntax

```
▶—QUIT—◀
```

### Parameters

None.

### Examples

#### Task

End an administrative client session in the interactive mode.

**Command:** QUIT

**Output:**

```
ANS5103I Highest return code was 0.
```

where *n* is the highest return code encountered during the administrative session.

### Related Commands

None.

## REDUCE DB

---

### REDUCE DB (Decrease the Assigned Capacity of the Database)

Use the REDUCE DB command to decrease the amount of space that can be used by the database. To reduce the capacity of the database, you must reduce the database in 4MB increments. If you do not specify the reduction in 4MB increments, ADSM rounds the number to the next 4MB partition. For example, if you reduce the database by 1MB, ADSM reduces the capacity of the database by 4MB.

**Attention:** The size of an allocated database, recovery log, or storage pool volume cannot be changed once it has been defined to the ADSM server. ADSM uses the initial size allocation of the volume at the time it is defined to the server to calculate data placement for later retrieval.

The REDUCE DB command can be initiated while users are accessing the server. Use the QUERY DB command to determine how much you can reduce the assigned capacity of the database.

This command can be used to free up a database volume for deletion. You can use the QUERY DBVOLUME command (FORMAT=DETAILED) to display the current utilization for your database volumes.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If a REDUCE DB background process is canceled, the database's assigned capacity may be partially reduced. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶▶ REDUCE DB nnn ▶▶
```

### Parameters

*nnn*

Specifies the number of megabytes to be removed from the assigned capacity of the database. This parameter is required. This value should be in increments of 4MB.

## REDUCE DB

### Examples

#### Task

Reduce the size of the database by 52MB.

**Command:** reduce db 52

#### Output:

```
ANR2250I Database assigned capacity has been reduced.
```

### Related Commands

Table 122 provides references to related commands.

Table 122. Commands Related to REDUCE DB

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE DBCOPY	Creates a volume copy of a database volume.	Page 116
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
EXTEND DB	Extends usable database storage within the volumes previously allocated for its use.	Page 269
QUERY DB	Displays allocation information about the database.	Page 339
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347
QUERY PROCESS	Displays information about one or all background server processes.	Page 419

## REDUCE LOG

---

### REDUCE LOG (Decrease the Assigned Capacity of the Recovery Log)

Use the REDUCE LOG command to decrease the amount of space that can be used by the recovery log. To decrease the capacity of the recovery log, you must reduce the recovery log in 4MB increments. If you do not specify the reduction in 4MB increments, ADSM rounds the number to the next 4MB partition. For example, if you reduce the recovery log by 1MB, ADSM reduces the capacity of the recovery log by 4MB.

The REDUCE LOG command can be initiated while users are accessing the server. Use the QUERY LOG command to determine how much you can reduce the assigned capacity of the recovery log.

The REDUCE LOG command can be used to free up a recovery log volume for deletion. You can use the QUERY LOGVOLUME command (FORMAT=DETAILED) to display the current utilization for your recovery log volumes.

This command can generate a background process that can be canceled with the CANCEL PROCESS command. If a REDUCE LOG background process is canceled, the recovery log's assigned capacity may be partially reduced. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
➤—REDUce LOG—nnn—————➤
```

### Parameters

*nnn*

Specifies the number of megabytes to be removed from the assigned capacity of the recovery log. This parameter is required. The value is in increments of 4MB.

### Examples

#### Task

Reduce the size of the recovery log by 16MB.

**Command:** reduce log 16

#### Output:

```
ANR2270I Recovery log assigned capacity has been reduced.
```

## REDUCE LOG

### Related Commands

Table 123 provides references to related commands.

*Table 123. Commands Related to REDUCE LOG*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
DEFINE LOGCOPY	Creates a volume copy of a recovery log volume.	Page 153
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
EXTEND LOG	Extends usable log storage within the volumes previously allocated for its use.	Page 271
QUERY DBBACKUPTRIGGER	Displays information about the database backup trigger.	Page 345
QUERY LOG	Displays allocation information about the recovery log.	Page 389
QUERY LOGVOLUME	Displays information about one or more log volumes.	Page 394
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514



## REGISTER ADMIN

---

### REGISTER ADMIN (Register an Administrator)

Use the REGISTER ADMIN command to add an administrator to the server. Once an administrator has been registered, the administrator can issue a limited set of commands, including all query commands. To provide additional administrator privileges, use the GRANT AUTHORITY command.

#### Privilege Class

To issue this command, you must have system privilege.

#### Syntax

```
▶▶ REGISTER Admin—adminname—password—[CONTACT=contactinfo]▶▶
```

#### Parameters

##### *adminname*

Specifies the name of the administrator to be registered. This parameter is required. The maximum length of the name is 64 characters.

##### *password*

Specifies the password of the administrator to be registered. This parameter is required. Use the SET PASSEXP command to specify how long an administrator or client node password is valid. The maximum length of the password is 64 characters.

##### **CONTACT**=*contactinfo*

Specifies information identifying the administrator being registered. This parameter is optional. The maximum length of this string is 255 characters. The contact information must be enclosed in quotation marks if it contains any blanks.

#### Examples

##### Task

Define an administrator, LARRY, with the password PASSONE. You can identify LARRY as second-shift personnel by specifying this information with the CONTACT parameter.

**Command:** register admin larry passone contact='second shift'

##### Output:

```
ANR2068I Administrator LARRY registered.
```

## REGISTER ADMIN

### Related Commands

Table 124 provides references to related commands.

*Table 124. Commands Related to REGISTER ADMIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
GRANT AUTHORITY	Assigns one or more privilege classes to an administrator.	Page 273
LOCK ADMIN	Prevents an administrator from accessing ADSM.	Page 303
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
REMOVE ADMIN	Removes an administrator from the list of registered administrators.	Page 470
RENAME ADMIN	Changes an ADSM administrator's name.	Page 474
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.	Page 521
UNLOCK ADMIN	Enables a locked administrator to access ADSM.	Page 535
UPDATE ADMIN	Changes the password or contact information associated with any administrator.	Page 537

## REGISTER LICENSE

---

### REGISTER LICENSE (Register a New License)

Use the REGISTER LICENSE command to register a new license with the ADSM server. Licenses are stored on the product CD-ROM in files called enrollment certificate files. These certificates are files that contain licensing information for the server product.

When registered, the licenses are then stored in a file named NODELOCK in the DSMSERV\_DIR environment variable, if specified, or in the current directory that the server was started from.

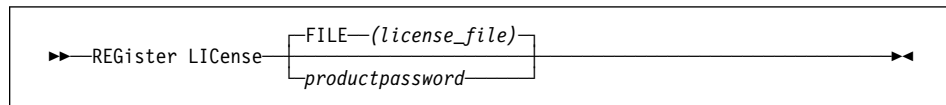
You can register new licenses to add:

- Additional clients
- Support for clients other than Windows NT
- Additional device support

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### FILE(*license\_file*)

Specifies the name of the enrollment certificate file containing the license to be registered. The specification can contain a filename wildcard (\*) For example, to register all files, enter:

```
register license FILE(*.LIC)
```

**Note:** The names of the files are not case-sensitive.

To register additional clients, specify the names of the following enrollment certificate files:

<b>1CLIENT.LIC</b>	1 backup-archive client
<b>5CLIENT.LIC</b>	5 backup-archive clients
<b>10CLIENT.LIC</b>	10 backup-archive clients
<b>50CLIENT.LIC</b>	50 backup-archive clients

To register clients other than OS/2, specify the names of the following enrollment certificate files:

## REGISTER LICENSE

<b>NETWORK.LIC</b>	License for Network connections. You must have this license if you will be using any communication method other than Named Pipes.
<b>DESKTOP.LIC</b>	Desktop clients (OS/2, Windows, Apple, Novell NetWare, and DOS)
<b>UNIX.LIC</b>	Any UNIX clients
<b>OEMVS.LIC</b>	The OpenEdition MVS client
<b>SPACEMAN.LIC</b>	HSM clients

To register device support modules, specify the names of the following enrollment certificate files:

<b>DEVMOD1.LIC</b>	Device Support Module 1
<b>DEVMOD2.LIC</b>	Device Support Module 2
<b>DEVMOD3.LIC</b>	Device Support Module 3
<b>DEVMOD4.LIC</b>	Device Support Module 4
<b>DEVM1T02.LIC</b>	Upgrade from Device Support Module 1 to 2
<b>DEVM2T03.LIC</b>	Upgrade from Device Support Module 2 to 3
<b>DEVM3T04.LIC</b>	Upgrade from Device Support Module 3 to 4

*productpassword*

Specifies the product password from the enrollment certificate for the selected feature to be registered.

## Examples

### Task

Register the DESKTOP clients.

**Command:** register license FILE(DESKTOP.LIC)

### Output:

```
ANR2835I REGister LICense 5s3qydpnw7njdxnafksqas4 DESKTOP
```

## Related Commands

Table 125 on page 465 provides references to related commands.

## REGISTER LICENSE

*Table 125. Commands Related to REGISTER LICENSE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LICENSES	Checks for compliance with defined licenses.	Page 54
QUERY LICENSE	Displays information about licenses and audits.	Page 386
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.	Page 512

## REGISTER NODE

---

### REGISTER NODE (Register a Client Node)

Use the REGISTER NODE command to register a client node to the server.

If your site has a closed registration system, then client nodes must be registered by an administrator with the REGISTER NODE command. You can use the REGISTER NODE command whether or not closed registration is in effect.

If your site is using open registration, client nodes can register themselves to the default policy domain named STANDARD, using only the default values for the attributes defined with the REGISTER NODE command.

**Note:** In a macro, the server may stall if some commands (for example, REGISTER NODE and DEFINE ASSOCIATION) are not committed as soon as you issue them. You could follow each command in a macro with a COMMIT command. However, a simpler solution is to include the -ITEMCOMMIT option with the DSMADMC command.

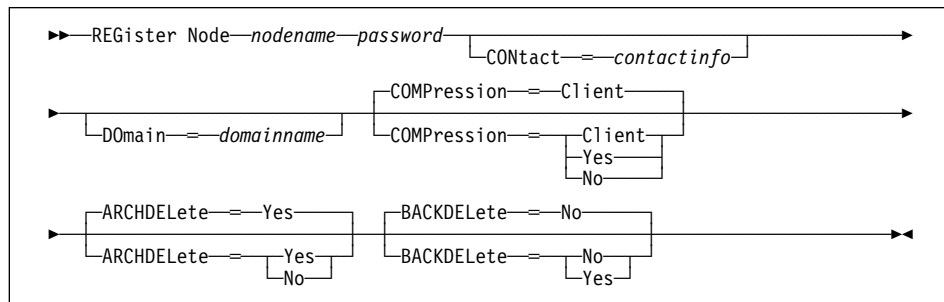
If a client requires a different policy domain than STANDARD, you must register the client node with this command.

The ARCHDELETE and BACKDELETE parameters determine whether the client node user can delete backup versions or archive copies from data storage. These parameters do not affect an administrator's ability to delete file spaces.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.

### Syntax



### Parameters

*nodename*

Specifies the name of the client node to be registered. This parameter is required. The maximum length of the name is 64 characters.

## REGISTER NODE

### *password*

Specifies the client node's password. This parameter is required. The maximum length of the name is 64 characters. Use the SET PASSEXP command to specify how long an administrator or client node password is valid.

### **CONTACT**=*contactinfo*

Specifies a text string of information identifying the node. This parameter is optional. The maximum length of the text string is 255 characters. The contact information must be enclosed in quotation marks if it contains any blanks.

### **DOMAIN**=*domainname*

Specifies the name of the policy domain to which the node is assigned. This parameter is optional. If you do not specify a policy domain name, the node is assigned to the default policy domain (STANDARD).

### **COMPRESSION**=*compressionvalue*

Specifies whether the client node compresses its files before sending them to the server for backup and archive. This parameter is optional. The default value is CLIENT. Possible values are:

#### **Client**

Specifies that the client determines whether to compress files.

#### **Yes**

Specifies that the client node compresses its files before sending them to the server for backup and archive.

**No** Specifies that the client node does not compress its files before sending them to the server for backup and archive.

### **ARCHDELETE**=*archdeletevalue*

Specifies whether the client node can delete its own archived files from the server. This parameter is optional. The default value is YES. Possible values are:

#### **Yes**

Specifies that the client node can delete its own archive files from the server.

**No** Specifies that the client node cannot delete its own archive files from the server.

### **BACKDELETE**=*backdeletevalue*

Specifies whether the client node can delete its own backup files from the server. This parameter is optional. The default value is NO. Possible values are:

**No** Specifies that the client node cannot delete its own backup files from the server.

#### **Yes**

Specifies that the client node can delete its own backup files from the server.

## REGISTER NODE

### Examples

#### Task

Register the client node JANET with the password SECRETCODE and assign this node to the EMPLOYEE\_RECORDS policy domain. This node can delete its own backup and archive files from the server. All files are compressed by the client node before they are sent to the server.

**Command:** register node janet secretcode archdelete=yes backdelete=yes  
compression=yes

#### Output:

```
ANR2060I Node JANET registered in policy domain EMPLOYEE_RECORDS.
```

### Related Commands

Table 126 provides references to related commands.

Command	Description	Reference
DELETE FILESPACE	Deletes data associated with one or more client's filespaces.	Page 216
LOCK NODE	Prevents a client from accessing the server.	Page 304
REMOVE NODE	Removes a client from the list of registered nodes for a specific policy domain.	Page 472
RENAME NODE	Changes the name for an existing user that belongs to a specific policy domain.	Page 479
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.	Page 521
SET REGISTRATION	Specifies whether users can register themselves or must be registered by an administrator.	Page 528



## REGISTER NODE

*Table 126 (Page 2 of 2). Commands Related to REGISTER NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
UNLOCK NODE	Enables a locked user in a specific policy domain to access the server.	Page 536
UPDATE NODE	Changes the attributes associated with a client node.	Page 585

## REMOVE ADMIN

---

### REMOVE ADMIN (Delete an Administrator)

Use the REMOVE ADMIN command to remove an administrator from the system.

You cannot remove the last system administrator from the system.

You cannot remove the SERVER\_CONSOLE administrative ID.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—REMOve Admin—adminname—◄◄
```

### Parameters

*adminname*

Specifies the name of the administrator to be removed. This parameter is required.

### Examples

#### Task

Remove the administrator LARRY.

**Command:** remove admin larry

#### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press ENTER key. The following is displayed:

```
ANR2069I Administrator LARRY removed.
```

## REMOVE ADMIN

### Related Commands

Table 127 provides references to related commands.

*Table 127. Commands Related to REMOVE ADMIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
LOCK ADMIN	Prevents an administrator from accessing ADSM.	Page 303
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
REGISTER ADMIN	Defines a new administrator without granting administrative authority.	Page 461
RENAME ADMIN	Changes an ADSM administrator's name.	Page 474

## REMOVE NODE

---

### REMOVE NODE (Delete a Client Node)

Use the REMOVE NODE command to remove a client node from the server.

Before you can remove a client node, you must delete all backup and archive file spaces that belong to that client node. To remove a client node, you should first use the QUERY SESSION command to determine if the node is in session. If necessary, use the CANCEL SESSION command to cancel the node's session. Use the LOCK NODE command to ensure that the client node does not start a new session and begin backup and archive operations while you are deleting the client node's file spaces. Use the QUERY FILESPACE command to determine the file spaces for the client node. Then use DELETE FILESPACE command to delete all file spaces for the client node. After all file spaces that belong to the client node are deleted, you can remove the client node from ADSM.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.

### Syntax

```
►►—REMove Node—nodename—————►►
```

### Parameters

*nodename*

Specifies the name of the client node to be removed. This parameter is required.

### Examples

#### Task

Remove the client node JANET.

**Command:** remove node janet

#### Output:

```
Do you wish to proceed? (Y/N)
```

Type Y and press the ENTER key. A message similar to the following is displayed:

```
ANR2061I Node JANET removed from policy domain STANDARD.
```

## REMOVE NODE

### Related Commands

Table 128 provides references to related commands.

*Table 128. Commands Related to REMOVE NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DELETE FILESPACE	Deletes data associated with one or more client's filespaces.	Page 216
LOCK NODE	Prevents a client from accessing the server.	Page 304
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403
QUERY SESSION	Displays information about all active administrator and client sessions with ADSM.	Page 427
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466
RENAME NODE	Changes the name for an existing user that belongs to a specific policy domain.	Page 479

## RENAME ADMIN

---

### RENAME ADMIN (Rename an Administrator)

Use the RENAME ADMIN command to change the name of an existing administrator. Existing information for this administrator such as password, contact information, and privilege classes is not altered.

If you are assigning an existing administrative user ID to another person, it is recommended that you use the UPDATE ADMIN command to change the password.

This command is useful for assigning an existing administrative user ID to another person who is not already an administrator. This command can also be used when an employee changes his or her name and wants to be identified by a new user ID.

You cannot rename the SERVER\_CONSOLE administrative ID.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—REName Admin—oldadminname—newadminname————►►
```

### Parameters

*oldadminname*

Specifies the name of the administrator to be renamed. This parameter is required.

*newadminname*

Specifies the new administrator name. This parameter is required. The maximum length of the name is 64 characters.

### Examples

#### Task

Rename the ADSM administrator CLAUDIA to BILL.

**Command:** rename admin claudia bill

#### Output:

```
ANR2070I Administrator CLAUDIA renamed to BILL.
```

## RENAME ADMIN

### Related Commands

Table 129 provides references to related commands.

*Table 129. Commands Related to RENAME ADMIN*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
UPDATE ADMIN	Changes the password or contact information associated with any administrator.	Page 537

## RENAME FILESPACE

---

### RENAME FILESPACE (Rename a Client Filespace on the Server)

Use the RENAME FILESPACE command to rename an existing client filespace on the server to a new filespace name.

This command is also useful after you have imported filespace into the server, prior to putting the filespace onto client nodes. Occasionally, the name of an imported filespace is the same as a filespace name that already exists on a client node. The server will not overlay an existing filespace that has the same name as an imported filespace. Instead, the imported filespace is given a newly system-generated filespace name. However, this new name may match a filespace name that already exists on the client node. (The client node may have filespace that have not yet been backed up and are unknown to the server.)

You can use the RENAME FILESPACE command to resolve this type of naming conflict or to rename the imported filespace to the naming convention used for the client node.

### Privilege Class

Any administrator with unrestricted policy authority or with restricted policy authority over the client's policy domain can issue this command.

### Syntax

```
►►—REName Filespace—nodename—filespace—newfilespace—◄◄
```

### Parameters

#### *nodename*

Specifies the name of the client node to which the filespace to be renamed belongs.

#### *filespace*

Specifies the name of the existing filespace defined for the client that is to be renamed.

A client filespace name is case sensitive; it must be specified using the same upper and lower case characters with which it is currently defined to the server.

#### *newfilespace*

Specifies the new name to be given to the client's filespace.

A client filespace name is case sensitive; it must be specified using the same upper and lower case characters with which it is to be defined to the server.

### Usage Notes

1. Use the QUERY FILESPACE command to display the names of the filespace defined for each client in the server database. You can use this command to make sure that you are renaming a filespace that is defined for the client in the server database.



## RENAME FILESPACE

2. After importing filespaces for a client node, this command can be used to rename the imported filespaces to more appropriate names for the client, or to avoid any naming conflicts with existing client naming conventions.

### Examples

#### Task

An AIX client node named LARRY has backed up filepace /r033 to the ADSM server. The filepace is exported to tape, then reimported to the server at a later time. When this filepace is imported, a system-generated name, r/031, is created for the imported filepace because /r033 already exists for client node LARRY.

Client node LARRY, however, already has a filepace named /r031 that has never been backed up and is unknown to the server. Unless the imported filepace is renamed, it will overlay filepace /r031 because the filepace name generated by the IMPORT function happened to be the same as a filepace on client node LARRY that was unknown to the server.

The confusion that this IMPORT operation introduces for client node LARRY is resolved by renaming imported filepace /r031 by using the following command:

**Command:** `rename filepace larry /r031 /imported-r033`

#### Output:

```
ANR0822I RENAME FILESPACE: Filespace r031 successfully
renamed imported-r033 for node LARRY
```

The new name, /imported-r033, clearly identifies that the new filepace is an imported image of filepace /r033.

### Related Commands

Table 130 provides references to related commands.

Table 130 (Page 1 of 2). Commands Related to RENAME FILESPACE

Command	Description	Reference
DELETE FILESPACE	Deletes data associated with one or more client's filespaces.	Page 216
EXPORT NODE	Copies client node information to external media.	Page 254

## RENAME FILESPACE

*Table 130 (Page 2 of 2). Commands Related to RENAME FILESPACE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY FILESPACE	Displays information about data in one or more filespaces that belong to a client.	Page 377
QUERY OCCUPANCY	Displays file space information by storage pool.	Page 408

## RENAME NODE

---

### RENAME NODE (Rename a Client Node)

Use the RENAME NODE command to rename a client node.

If you are assigning an existing client node ID to another person, it is recommended that you use the UPDATE NODE command to change the password.

This command is useful for assigning an existing client node user ID to another person. This command can also be used when an employee changes his or her name and wants to be identified by a new user ID.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.

### Syntax

```
►►—REName Node—oldnodename—newnodename—◄◄
```

### Parameters

*oldnodename*

Specifies the name of the client node to be renamed. This parameter is required.

*newnodename*

Specifies the new name of the client node. This parameter is required. The maximum length of the name is 64 characters.

### Examples

#### Task

Rename the client node JOE to JOYCE.

**Command:** rename node joe joyce

#### Output:

```
ANR2062I Node JOE renamed to JOYCE.
```

## RENAME NODE

### Related Commands

Table 131 provides references to related commands.

*Table 131. Commands Related to RENAME NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403
UPDATE NODE	Changes the attributes associated with a client node.	Page 585

## REPLY

---

### REPLY (Allow a Request to Continue Processing)

Use the REPLY command to signal completion of an operation and for the server to invoke the next operation. To initiate such activity, the server issues a request message that contains an identification number. Use the REPLY command and the identification number to inform the server that you have completed the requested operation. Not all requests require a reply; this command is only required if the request message specifically indicates that a REPLY is needed.

#### Privilege Class

To issue this command, you must have system privilege or operator privilege.

#### Syntax

```
►►—REPLY—requestnum—◄◄
```

#### Parameters

*requestnum*

Specifies the identification number of the request that has been completed. This parameter is required.

#### Examples

##### Task

Respond to a reply request using 3 as the request number.

**Command:** reply 3

##### Output:

```
ANR8499I Command accepted.
```

#### Related Commands

Table 132 provides references to related commands.

Table 132. Commands Related to REPLY

Command	Description	Reference
CANCEL REQUEST	Cancels one or more pending volume mount requests.	Page 76
QUERY REQUEST	Displays information about all pending mount requests.	Page 421

## RESET BUFPOOL

---

### RESET BUFPOOL (Reset the Buffer Pool Statistics for the Database)

Use the RESET BUFPOOL command to reset the buffer pool statistics for the database. ADSM does not issue a response when you enter this command. The buffer pool statistics consist of:

- The cumulative number of database requests since the last RESET BUFPOOL command (or since the server was last started)
- The percentage of database requests for database pages that are already in the database buffer pool (as cached copies)
- The percentage of database requests that had to wait for a page in the database buffer pool to become available.

You can use the QUERY DB command (FORMAT=DETAILED) to display the current values for the buffer pool statistics.

### Privilege Class

To issue this command, you must have system privilege or analyst privilege.

### Syntax

```
▶▶—RESet BUFPool—————▶▶
```

### Parameters

None.

### Examples

#### Task

Reset the buffer pool statistics.

**Command:** reset bufpool

#### Output:

```
(no response)
```

## RESET BUFPOOL

### Related Commands

Table 133 provides references to related commands.

*Table 133. Commands Related to RESET BUFPOOL*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY DB	Displays allocation information about the database.	Page 339
RESET DBMAXUTILIZATION	Resets the maximum utilization statistic for the database.	Page 484
RESET LOGMAXUTILIZATION	Resets the maximum utilization statistic for the recovery log.	Page 487

## RESET DBMAXUTILIZATION

---

### RESET DBMAXUTILIZATION (Reset the Utilization Statistic for the Database)

Use the RESET DBMAXUTILIZATION command to reset the maximum utilization statistic for the database. ADSM does not issue a response when you enter this command.

This command resets the value of the maximum utilization statistic for the database to the current utilization. You can display the current value of the maximum utilization statistic for the database with the QUERY DB command.

#### Privilege Class

To issue this command, you must have system privilege or analyst privilege.

#### Syntax

```
▶▶—RESet DBMaxutilization—————▶▶
```

#### Parameters

None.

#### Examples

##### Task

Reset the maximum utilization statistic for the database.

**Command:** reset dbmaxutilization

##### Output:

```
(no response)
```

#### Related Commands

Table 134 provides references to related commands.

*Table 134. Commands Related to RESET DBMAXUTILIZATION*

Command	Description	Reference
QUERY DB	Displays allocation information about the database.	Page 339
RESET BUFPOOL	Resets the buffer pool statistics for the database.	Page 482



## RESET LOGCONSUMPTION

---

### RESET LOGCONSUMPTION (Reset the Cumulative Log Consumption Statistic)

Use the RESET LOGCONSUMPTION command to reset the statistic on the amount of recovery log space that has been consumed since the last reset.

You can run this command at any time.

This command allows an administrator to reset to zero the counter of megabytes of recovery log that have been consumed since the last reset. The recovery log consumption statistic is a cumulative measure of the amount of log space used by the server.

#### Privilege Class

To issue this command, you must have analyst privilege.

#### Syntax

```
▶▶—RESet LOGConsumption—▶▶
```

#### Parameters

None.

#### Examples

##### Task

Reset the statistic for recovery log consumption.

**Command:** reset logconsumption

##### Output:

```
(No response)
```

#### Related Commands

Table 135 provides references to related commands.

---

*Table 135 (Page 1 of 2). Commands Related to RESET LOGCONSUMPTION*

Command	Description	Reference
QUERY LOG	Displays allocation information about the recovery log.	Page 389

---

## RESET LOGCONSUMPTION

*Table 135 (Page 2 of 2). Commands Related to RESET LOGCONSUMPTION*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514

## RESET LOGMAXUTILIZATION

---

### RESET LOGMAXUTILIZATION (Reset the Utilization Statistic for the Log)

Use the RESET LOGMAXUTILIZATION command to reset the maximum utilization statistic for the recovery log. ADSM does not issue a response when you enter this command.

This command resets the value for the maximum utilization statistic of the recovery log to 0. You can display the current value of the maximum utilization statistic for the recovery log with the QUERY LOG command.

#### Privilege Class

To issue this command, you must have system privilege or analyst privilege.

#### Syntax

```
▶▶—RESet LOGMaxutilization—————▶▶
```

#### Parameters

None.

#### Examples

##### Task

Reset the maximum utilization statistic for the recovery log.

**Command:** reset logmaxutilization

##### Output:

```
(no response)
```

#### Related Commands

Table 136 provides a reference to a related command.

*Table 136. Command Related to RESET LOGMAXUTILIZATION*

Command	Description	Reference
QUERY LOG	Displays allocation information about the recovery log.	Page 389

## RESTORE STGPOOL

---

### RESTORE STGPOOL (Restore Storage Pool Data from a Copy Pool)

Use the RESTORE STGPOOL command to restore files from one or more copy storage pools to a primary storage pool. The files must have been copied to the copy storage pools by using the BACKUP STGPOOL command.

When you use this command, ADSM restores all files that belong to the specified primary storage pool and that meet one of the following criteria:

- The primary copy of the file has been identified as having data-integrity errors during a previous operation.
- The primary copy of the file resides on a volume that has an access mode of DESTROYED.

After the files are restored, the old references to these files in the primary storage pool are deleted from the database. ADSM now locates these files on the volumes to which they were restored, rather than on the volumes on which they were previously stored. If a destroyed volume becomes empty because all files have been restored to other locations, the destroyed volume is automatically deleted from the database.

**Note:** Cached copies of files are never restored. Any cached files that have data-integrity errors or that reside on a *destroyed* volume are removed from the database during restore processing.

The restoration may be incomplete for one or more of the following reasons:

- Files were either never backed up or the backup copies are marked as damaged. Use the QUERY CONTENT command to get more information on the remaining files on the volume.
- A copy storage pool was specified on the RESTORE command, but files were backed up to a different copy storage pool. Use the PREVIEW parameter when you issue the RESTORE command again to determine if this is the problem.
- Volumes in the copy storage pool needed to perform the restore operation are offsite or unavailable. Check the activity log for messages that occurred during restore processing.
- Backup file copies in copy storage pools were moved or deleted by other ADSM processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes while restore processing is in progress:
  - MOVE DATA
  - DELETE VOLUME (DISCARDDATA=YES)
  - AUDIT VOLUME (FIX=YES)

Also, you can prevent reclamation processing for your copy storage pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

This command can be used to identify volumes that contain damaged, primary files. During restore processing, a message is issued for every volume in the restored

# RESTORE STGPOOL

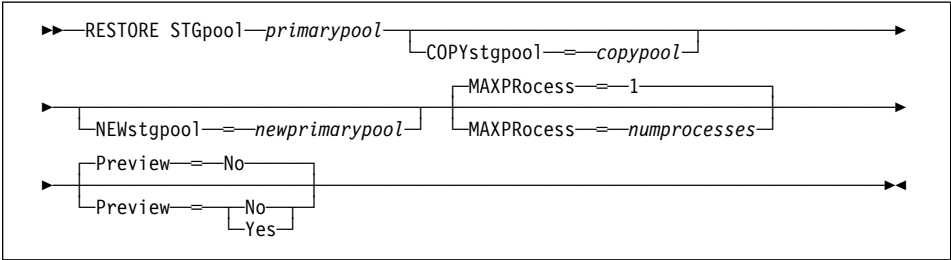
storage pool that contains damaged, non-cached files. Use the QUERY CONTENT command to identify damaged, primary files on a specific volume.

This command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE STGPOOL background process is canceled, some files may have already been restored prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

## Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the primary storage pool for which files are to be restored. If you are a restricted storage administrator and you want to restore files to a new primary storage pool, you must also have authority for the new storage pool.

## Syntax



## Parameters

**primarypool**  
Specifies the name of the primary storage pool that is being restored. This parameter is required.

**COPYstgpool=copypool**  
Specifies the name of the copy storage pool from which the files are to be restored. This parameter is optional. If this parameter is not specified, files are restored from any copy pool in which copies can be located.

**NEWstgpool=newprimarypool**  
Specifies the name of the new storage pool to which to restore the files. This parameter is optional. If this parameter is not specified, files are restored to the original primary storage pool (the pool being restored).

**MAXProcess=numprocesses**  
Specifies the maximum number of parallel processes that are used for restoring files. By using multiple, parallel processes, you may be able to improve throughput for the restore operation. This parameter is optional. You can specify a value from 1–999, inclusive. The default value is 1.

When determining the number of processes to use, consider the number of mount points and drives that can be dedicated to this operation. To access a volume in a sequential access device class, ADSM uses a mount point (a logical

## RESTORE STGPOOL

drive), and, if the sequential device class is not FILE, a physical drive. The number of mount points and drives available depends on other ADSM and system activity, and on the mount limits of the device classes for the storage pools that are involved in the restore operation. The maximum number of mount points for a device class is specified with the mount limit parameter, which is the number of logical drives that ADSM can use at one time to access volumes in that device class.

Each process needs a mount point for copy storage pool volumes, and, if the device class is not FILE, each process also needs a drive. If the storage pool to which you are restoring files is a sequential storage pool, each process needs an additional mount point for primary storage pool volumes, and, if the device class is not FILE, an additional drive.

For example, suppose you specify MAXPROCESS=3 to restore files to a primary sequential storage pool from a copy storage pool, both assigned to the same 8MM device class. Each of the three processes requires two mount points and two drives. For all three processes to run, the 8MM device class must have a mount limit of no less than 6, and there must be at least 6 mount points and 6 drives currently available.

If you specify PREVIEW=YES, only one process is used, regardless of the value specified for the MAXPROCESS parameter. No mount points or drives are required when you specify PREVIEW=YES.

### **Preview=***value*

Specifies whether you only want to preview the restore operation without actually performing it. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that you want to perform the restore operation.

#### **Yes**

Specifies that you only want to preview the restore operation, without actually restoring the data. This option can be used to identify offsite volumes that would be required to restore a storage pool. This option displays:

- A list of volumes in the specified primary storage pool that contain damaged, primary files.
- The number of files and the number of bytes that will be restored, assuming that the access mode of the required copy storage pool volumes is READWRITE or READONLY when the restore operation is performed.
- A list of copy storage pool volumes containing files that will be restored. These volumes must be mounted if you actually perform the restore operation.
- A list of any volumes containing files that cannot be restored.

## RESTORE STGPOOL

**Note:** If you preview a restore operation to obtain a list of offsite copy pool volumes that must be mounted, it is recommended that you change the access mode of the identified copy pool volumes to UNAVAILABLE. This prevents reclamation and MOVE DATA processing for these volumes until they have been transported to the onsite location for use in restore processing.

### Examples

#### Task

Restore files from any copy storage pool to the primary storage pool, PRIMARY\_POOL.

**Command:** restore stgpool primary\_pool

#### Output:

```
ANR1230I Restore of primary storage pool PRIMARY_POOL started
as process 18.
ANR2110I RESTORE STGPOOL started as process 18.
ANR1254I Removable volume c:\admserv\mnvol is required
for data storage restore.
ANR8324I FILE volume c:\admserv\mnvol is expected to be
mounted (R/W).
adsm>
ANR8340I FILE volume c:\admserv\data2.dsm mounted.
ANR8340I FILE volume c:\admserv\mnvol mounted.
ANR1234I Restore process 18 ended for storage pool PRIMARY_POOL.
ANR1238I Restore of primary storage pool PRIMARY_POOL has ended.
Files Restored: 42, Bytes Restored: 7452550, Unreadable Files: 0,
Unreadable Bytes: 0.
ANR2208I Volume c:\admserv\data1.dsm deleted from storage pool
PRIMARY_POOL.
```

### Related Commands

Table 137 provides references to related commands.

Table 137 (Page 1 of 2). Commands Related to RESTORE STGPOOL

Command	Description	Reference
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
CANCEL PROCESS	Cancels a background server process.	Page 74
QUERY CONTENT	Displays information about one or more files currently residing in a storage pool volume.	Page 328

## RESTORE STGPOOL

*Table 137 (Page 2 of 2). Commands Related to RESTORE STGPOOL*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
RESTORE VOLUME	Restores files stored on specified volumes in a primary storage pool from copy storage pools.	Page 493
UPDATE STGPOOL	Changes the attributes of a storage pool.	Page 605



## RESTORE VOLUME

---

### RESTORE VOLUME (Restore Primary Volume Data from a Copy Pool)

Use the RESTORE VOLUME command to restore all files that are currently stored on one or more volumes in the same primary storage pool, and that were previously backed up to copy storage pools by using the BACKUP STGPOOL command. The files are restored from one or more copy storage pools. You can use this command to restore files on volumes known to be damaged, for example, a broken 8mm tape.

If more than one volume is specified to be restored, this command attempts to minimize volume mounts for the copy storage pool. Therefore, to restore more than one volume in the same primary storage pool, issue this command once and specify a list of volumes to be restored.

**Note:** Cached copies of files are never restored. Any cached files that reside on a volume that is being restored are removed from the database during restore processing.

Before you can use this command to restore a random-access volume, you must use the VARY command to vary the volume offline.

After files are restored, the old references to these files in the primary storage pool are deleted from the database. ADSM will now locate these files on the volumes to which they were restored, rather than on the volume on which they were previously stored.

This command changes the access mode of the specified volumes to DESTROYED. When the restoration is complete (all files on the volume are restored to other locations), the destroyed volume is empty and is then automatically deleted from the database.

The restoration may be incomplete for one or more of the following reasons:

- Files were either never backed up or the backup copies are marked as damaged. Use the QUERY CONTENT command to get more information on the remaining files on the volume.
- A copy storage pool was specified on the RESTORE command, but files were backed up to a different copy storage pool. Use the PREVIEW parameter when you issue the RESTORE command again to determine if this is the problem.
- Volumes in the copy storage pool needed to perform the restore operation are offsite or unavailable. Check the activity log for messages that occurred during restore processing.
- Backup file copies in copy storage pools were moved or deleted by other ADSM processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool volumes while restore processing is in progress:
  - MOVE DATA
  - DELETE VOLUME (DISCARDATA=YES)
  - AUDIT VOLUME (FIX=YES)

## RESTORE VOLUME

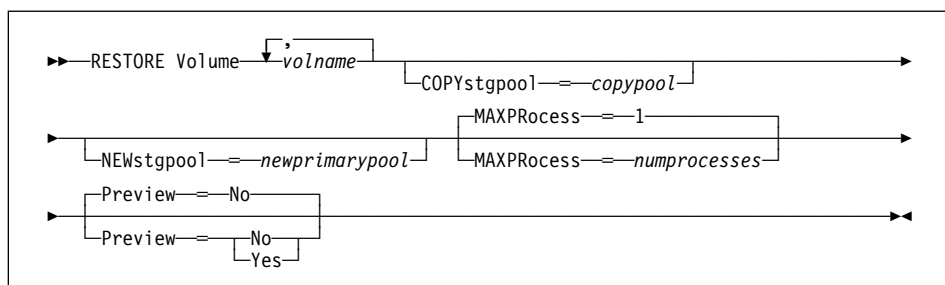
Also, you can prevent reclamation processing for your copy storage pools by setting the RECLAIM parameter to 100 with the UPDATE STGPOOL command.

This command generates a background process that can be canceled with the CANCEL PROCESS command. If a RESTORE VOLUME background process is canceled, some files may have already been restored prior to the cancellation. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the primary storage pool to which the specified volumes belong. If you are a restricted storage administrator and you want to restore files to a new primary storage pool, you must also have authority for the new storage pool.

### Syntax



### Parameters

#### *volname*

Specifies the name of the volume in a primary storage pool for which to restore files. This parameter is required. You can specify a list of volumes, if all volumes in the list belong to the same primary storage pool. The items in the list are separated by commas, with no intervening spaces.

#### **COPYstgpool**=*copypool*

Specifies the name of the copy storage pool from which the files are to be restored. This parameter is optional. If this parameter is not specified, files are restored from any copy pool in which copies can be located.

#### **NEWstgpool**=*newprimarypool*

Specifies the name of the new storage pool to which to restore the files. This parameter is optional. If this parameter is not specified, files are restored to the primary storage pool to which the volume or volumes being restored belong.

#### **MAXPRocess**=*numprocesses*

Specifies the maximum number of parallel processes that are used for restoring files. By using multiple, parallel processes, you may be able to improve throughput for the restore operation. This parameter is optional. You can specify a value from 1–999, inclusive. The default value is 1.

## RESTORE VOLUME

When determining the number of processes to use, consider the number of mount points and drives that can be dedicated to this operation. To access a volume in a sequential access device class, ADSM uses a mount point (a logical drive), and, if the sequential device class is not FILE, a physical drive. The number of mount points and drives available depends on other ADSM and system activity, and on the mount limits of the device classes for the storage pools that are involved in the restore operation. The maximum number of mount points for a device class is specified with the mount limit parameter, which is the number of logical drives that ADSM can use at one time to access volumes in that device class.

Each process needs a mount point for copy storage pool volumes, and, if the device class is not FILE, each process also needs a drive. If the storage pool to which you are restoring files is a sequential storage pool, each process needs an additional mount point for primary storage pool volumes, and, if the device class is not FILE, an additional drive.

For example, suppose you specify `MAXPROCESS=3` to restore files to a primary sequential storage pool from a copy storage pool, both assigned to the same 8MM device class. Each of the three processes requires two mount points and two drives. For all three processes to run, the 8MM device class must have a mount limit of no less than 6, and there must be at least 6 mount points and 6 drives currently available.

If you specify `PREVIEW=YES`, only one process is used, regardless of the value specified for the `MAXPROCESS` parameter. No mount points or drives are required when you specify `PREVIEW=YES`.

### **Preview=value**

Specifies whether you want to preview the restore operation without actually restoring data. This parameter is optional. The default value is `NO`. Possible values are:

#### **No**

Specifies that you want to perform the restore operation.

#### **Yes**

Specifies that you want to preview the restore operation without actually restoring the data. This option displays:

- The number of files and the number of bytes that will be restored, assuming that the access mode of the required copy storage pool volumes is `READWRITE` or `READONLY` when the restore operation is performed.
- A list of copy storage pool volumes containing files that will be restored. These are volumes that must be mounted if you actually perform the restore.
- A list of any volumes containing files that are not restorable.

This option can be used to identify offsite volumes that would be required to restore a storage pool.

## RESTORE VOLUME

**Note:** If you preview this restore operation to obtain a list of offsite copy pool volumes that must be mounted, it is recommended that you change the access mode of the identified copy pool volumes to UNAVAILABLE. This prevents reclamation and MOVE DATA processing for these volumes until they have been transported to the onsite location for use in restore processing.

### Examples

#### Task

Restore files stored on volume PVOL2 in primary storage pool PRIMARY\_POOL.

**Command:** restore volume pvo12

#### Output:

```
ANR2114I RESTORE VOLUME: Access mode for volume c:\admserv\data1.dsm
updated to "destroyed".
ANR1232I Restore of volumes in primary storage pool PRIMARY_POOL
started as process 19.
ANR1254I Removable volume c:\admserv\mnvol is required for data
storage restore.
ANR2110I RESTORE VOLUME started as process 19.
ANR8324I FILE volume c:\admserv\mnvol is expected to be mounted
(R/W).
adsm>
ANR8340I FILE volume c:\admserv\data2.dsm mounted.
ANR8340I FILE volume c:\admserv\mnvol mounted.
ANR1235I Restore process 19 ended for volumes in storage pool
PRIMARY_POOL.
ANR1240I Restore of volumes in primary storage pool PRIMARY_POOL
has ended.
Files Restored: 42, Bytes Restored: 7452550, Unreadable Files: 0,
Unreadable Bytes: 0.
ANR2208I Volume c:\admserv\data1.dsm deleted from storage pool
PRIMARY_POOL.
```

## RESTORE VOLUME

### Related Commands

Table 138 provides references to related commands.

*Table 138. Commands Related to RESTORE VOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
CANCEL PROCESS	Cancels a background server process.	Page 74
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
RESTORE STGPOOL	Restores files to a primary storage pool from copy storage pools.	Page 488

## REVOKE AUTHORITY

### REVOKE AUTHORITY (Remove Administrator Authority)

Use the REVOKE AUTHORITY command to revoke one or more privilege classes from an administrator. You can also use this command to reduce the number of policy domains to which a restricted policy administrator has authority. This command can also be used to reduce the number of storage pools to which a restricted storage administrator has authority.

At least one administrator must have system privilege; therefore, if the administrator is the only one with system privilege, you cannot revoke his or her authority.

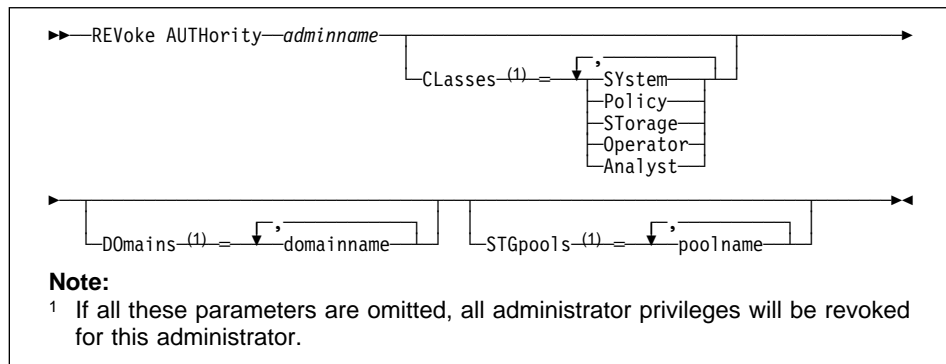
To change an unrestricted policy administrator to a restricted policy administrator, you must first use this command to revoke the unrestricted privilege. Then, use the GRANT AUTHORITY command to grant the administrator restricted policy privilege and to identify the policy domains to which the administrator has authority.

To change an unrestricted storage administrator to a restricted storage administrator, you must first use this command to revoke the unrestricted privilege. Then, use the GRANT AUTHORITY command to grant the administrator restricted storage privilege and to identify the storage pools to which the administrator has authority.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

*adminname*

Specifies the name of the administrator whose administrative privilege is to be revoked. This parameter is required.

**CLASSES**=*classes*

Specifies one or more administrative privilege classes to be revoked. You can specify more than one class by delimiting each with a comma.

## REVOKE AUTHORITY

### **S**ystem

Indicates that system authority is to be revoked for this administrator. If CLASSES=SYSTEM is specified, no other classes can be specified, and the DOMAINS and STGPools parameters cannot be specified.

### **P**olicy

Indicates that policy privilege is to be revoked for this administrator. To revoke all policy privilege, specify CLASSES=POLICY and do not specify the DOMAINS parameter.

### **S**torage

Indicates that storage privilege is to be revoked for this administrator. To revoke all storage privilege, specify CLASSES=STORAGE and do not specify the STGPools parameter.

### **O**perator

Indicates that operator privilege is to be revoked for this administrator.

### **A**nalyst

Indicates that analyst privilege is to be revoked for this administrator.

### **DO**main*s=domainlist*

Specifies a list of policy domains that can no longer be managed by a restricted policy administrator. (The administrator had been authorized to manage these domains until the REVOKE command was issued.) This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use a pattern-matching expression to specify a name. Authority for all matching domains will be revoked. If DOMAINS is specified, the parameter CLASSES=POLICY is optional.

### **STG**pool*s=poollist*

Specifies a list of storage pools that can no longer be managed by a restricted policy administrator. (The administrator had been authorized to manage these storage pools until the REVOKE command was issued.) This parameter is optional. The items in the list are separated by commas, with no intervening spaces. You can use a pattern-matching expression to specify a name. Authority for all matching storage pools will be revoked. If STGPools is specified then the parameter CLASSES=STORAGE is optional.

## Usage Notes

1. If you use the REVOKE AUTHORITY command without the CLASSES, DOMAINS, and STGPools parameters, you will revoke all privileges for the specified administrator.
2. You can revoke unrestricted policy privilege from an administrator by specifying the CLASSES=POLICY parameter. You cannot use the DOMAINS parameter to revoke authority for selected domains from an unrestricted policy administrator.
3. You can revoke unrestricted storage privilege from an administrator by specifying the CLASSES=STORAGE parameter. You cannot use the STGPools parameter to revoke authority for selected storage pools from an unrestricted storage administrator.

## REVOKE AUTHORITY

### Examples

#### Task 1

Revoke part of administrator CLAUDIA's privileges. CLAUDIA has restricted policy privilege for the policy domains EMPLOYEE\_RECORDS and PROG1. Restrict CLAUDIA's policy privilege to the EMPLOYEE\_RECORDS policy domain.

**Command:** `revoke authority claudia classes=policy domains=employee_records`

#### Output:

```
ANR2085I Restricted policy privilege revoked for
administrator CLAUDIA - policy domain PROG1.
ANR2085I Restricted policy privilege revoked for
administrator CLAUDIA - policy domain EMPLOYEE_RECORDS.
```

#### Task 2

Administrator LARRY currently has operator, analyst, and restricted policy privilege. Revoke all administrative privileges for administrator LARRY. To revoke all administrative privileges for an administrator, identify the administrator, but do not specify CLASSES, DOMAINS, or STGPOLLS. LARRY remains an administrator but he can only use those commands that can be issued by any administrator.

**Command:** `revoke authority larry`

#### Output:

```
ANR2085I Restricted policy privilege revoked for
administrator LARRY - policy domain PROG1.
ANR2088I Analyst privilege revoked for administrator LARRY.
ANR2089I Operator privilege revoked for administrator LARRY.
```

#### Task 3

Revoke system privilege for administrator TOM. TOM remains an administrator but he can only use those commands that can be issued by any administrator. (You can, if needed, use the GRANT AUTHORITY command to give privileges to TOM.)

**Command:** `revoke authority tom classes=system`

#### Output:

```
ANR2083I System privilege revoked for administrator TOM.
```



## REVOKE AUTHORITY

### Task 4

Administrator JANET currently has operator, analyst, and unrestricted storage privilege. Revoke her operator and analyst privileges. After the command is issued, the administrator JANET continues to have unrestricted storage privilege.

**Command:** `revoke authority janet classes=operator,analyst`

**Output:**

```
ANR2088I Analyst privilege revoked for administrator JANET.  
ANR2089I Operator privilege revoked for administrator JANET.
```

## REVOKE AUTHORITY

### Task 5

Administrator JANE currently has operator, analyst, and unrestricted storage privilege. Revoke her operator and analyst privileges. After the command is issued, the administrator JANE continues to have unrestricted storage privilege.

**Command:** `revoke authority jane classes=operator,analyst`

**Output:**

```
ANR2088I Analyst privilege revoked for administrator JANE.  
ANR2089I Operator privilege revoked for administrator JANE.
```

### Related Commands

Table 139 provides references to related commands.

*Table 139. Commands Related to REVOKE AUTHORITY*

Command	Description	Reference
GRANT AUTHORITY	Assigns one or more privilege classes to an administrator.	Page 273
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317

## ROLLBACK

---

### ROLLBACK (Rollback Uncommitted Changes in a Macro)

Use the ROLLBACK command within a macro to undo any processing changes made by commands that have been executed by the server but not yet committed to the database. After a change has been committed, it is permanent; it cannot be rolled back. This command is useful for testing macros.

Make sure that your administrative client session is not running with the ITEMCOMMIT option when using this command. For more information on controlling command processing in a macro, see “Controlling Command Processing in a Macro” on page 25.

#### Privilege Class

Any administrator can issue this command.

#### Syntax

```
▶▶—ROLLBACK—◀◀
```

#### Parameters

None.

#### Examples

##### Task

Run the REGN macro with the ROLLBACK command to verify that the macro works without committing any changes.

##### Macro Contents:

```
/* Macro to register policy
administrators & grant authority */
REGister Admin sara hobby
GRant AUTHority sara CLasses=Policy
REGister Admin ken plane
GRant AUTHority ken CLasses=Policy
ROLLBACK /* prevents any changes from being committed */
```

**Command:** macro regn

**Output:**

## ROLLBACK

```
ANS5101I Server command: 'REGister Admin sara hobby'  
ANR2068I Administrator SARA registered.  
  
ANS5101I Server command: 'GRant AUTHority sara  
CLasses=policy'  
ANR2077I Unrestricted policy privilege granted to  
administrator SARA.  
  
ANS5101I Server command: 'REGister Admin ken plane'  
ANR2068I Administrator KEN registered.  
  
ANS5101I Server command: 'GRant AUTHority ken CLasses=policy'  
ANR2077I Unrestricted policy privilege granted to  
administrator KEN.  
  
ANS5101I Server command: 'ROLLBACK'
```

### Related Commands

Table 140 provides references to related commands.

*Table 140. Commands Related to ROLLBACK*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
COMMIT	Makes changes to the database permanent.	Page 87
MACRO	Runs a specified macro file.	Page 306

## SET ACCOUNTING

---

### SET ACCOUNTING (Set Accounting Records On or Off)

Use the SET ACCOUNTING command to determine whether an accounting record is created every time a client node session ends.

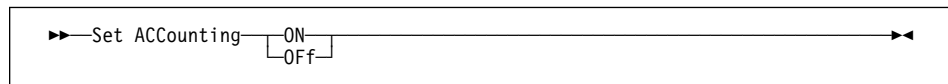
An accounting record tracks the amount of storage used by a client node session. Accounting records are stored in an accounting file named dsmacct.log which is located in the current directory when the server is started (it is not located in the directory pointed to by the DSMSEV\_DIR environment variable). This allows multiple servers on the same system to maintain separate accounting logs. The file is written out as text records that can be viewed directly or can easily be read into a spreadsheet, such as Lotus 123. Individual fields are delimited by commas (,) and records are terminated with the newline character. See the *ADSM Administrator's Guide* for more information on accounting.

Use the QUERY STATUS command to determine whether accounting records are generated. At installation, this value is set to off.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### ON

Specifies that the server creates an accounting record every time a client node session ends.

#### OFF

Specifies that the server does not create accounting records.

### Examples

#### Task

Create an accounting record at the end of each client node session.

**Command:** set accounting on

#### Output:

```
ANR2091I Accounting parameter set to ON.
```

## SET ACCOUNTING

### Related Commands

Table 141 provides references to related commands.

---

*Table 141. Commands Related to SET ACCOUNTING*

---

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

---

## SET ACTLOGRETENTION

---

### SET ACTLOGRETENTION (Set the Retention Period for the Activity Log)

Use the SET ACTLOGRETENTION command to specify the retention period, in days, for server activity log records.

This command allows you to define how long activity log information is retained in the database. Activity log information includes messages, such as:

- When client sessions start and end
- When migration starts and ends
- Diagnostic error messages
- Scheduled administrative command output

Because the activity log is stored in the database, the size of the activity log should be factored into the amount of space allocated for the database. The activity log size is dependent upon the length of time the activity log information is retained. When the length of time is increased, the amount of accumulated data also increases, so that additional database storage is required. When sufficient space is not available for activity log records, ADSM suspends recording, and sends messages about this condition to the server console. If you increase the size of the database, ADSM starts activity log recording again.

After a message has passed the specified retention period, the server automatically deletes it.

Use the QUERY ACTLOG command to search for messages in the activity log.

Use the QUERY STATUS command to display the current retention period for the activity log. At installation, this value is set to 1 day.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set ACTLogretention—days—◄◄
```

### Parameters

*days*

Specifies an integer that defines the number of days to retain messages in the activity log. This parameter is required. You can specify an integer in the range of 0 to 9999, inclusive. A value of 0 disables activity log retention. A value of 1 specifies to only retain the activity log records for the current day.

## SET ACTLOGRETENTION

### Examples

#### Task

Set the server to retain activity log records for 30 days.

**Command:** set actlogretention 30

#### Output:

```
ANR2090I Activity log retention period set to 30 days.
```

### Related Commands

Table 142 provides a reference to a related command.

*Table 142. Command Related to SET ACTLOGRETENTION*

Command	Description	Reference
QUERY ACTLOG	Searches for and displays messages from the server activity log.	Page 311



## SET AUTHENTICATION

---

### SET AUTHENTICATION (Set Password Authentication)

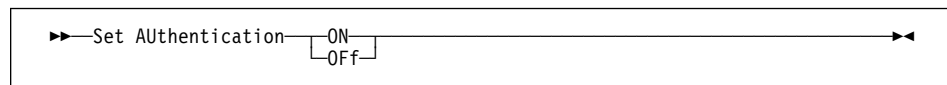
Use the SET AUTHENTICATION command to specify whether administrators and client nodes must use a password to access the server.

Use the QUERY STATUS command to display the current authentication setting. At installation, this value is set to on.

#### Privilege Class

To issue this command, you must have system privilege.

#### Syntax



#### Parameters

##### ON

Specifies that administrators and client nodes need a password to access the server.

##### OFF

Specifies that administrators and client nodes do not need a password to access the server.

#### Examples

##### Task

Set authentication to ON to require administrators and client nodes to enter a password when accessing the server.

**Command:** `set authentication on`

##### Output:

```
ANR2095I Authentication parameter set to ON.
```

#### Related Commands

Table 143 provides a reference to a related command.

Table 143. Command Related to SET AUTHENTICATION

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## SET EVENTRETENTION

---

### SET EVENTRETENTION (Specify the Retention Period for Event Records)

Use the SET EVENTRETENTION command to specify the retention period for event records in the server database. An event record is created whenever processing of a scheduled command is started or missed.

Event records are automatically removed from the database once both of the following conditions have been met:

- The specified retention period has passed from the scheduled execution date and time for the event
- The startup window for the event has elapsed.

Use this command to set a retention period for event records that will permit you to effectively monitor completed schedules. You can adjust the length of time event information is maintained to avoid insufficient or outdated data. To display event records, use the QUERY EVENT command.

You can use the DELETE EVENT command to delete event records regardless of whether their retention period has elapsed.

You can display the value for the event retention period with the QUERY STATUS command. At installation, this value is set to 10 days.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set Eventretention—days—◄◄
```

### Parameters

*days*

Specifies the number of days for which event records are retained in the database. This parameter is required. You can specify an integer in the range of 0 to 9999, inclusive. A value of 0 indicates that only event records for the current day will be retained.

### Examples

#### Task

Maintain event records in the database for a period no longer than 15 days.

**Command:** set eventretention 15

## SET EVENTRETENTION

### Output:

```
ANR2521I Event record retention period set to 15 day(s).
```

### Related Commands

Table 144 provides references to related commands.

*Table 144. Commands Related to SET EVENTRETENTION*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DELETE EVENT	Deletes event records prior to a specified date and time.	Page 212
QUERY EVENT	Displays information about scheduled and completed events for selected clients.	Page 362
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## SET LICENSEAUDITPERIOD

---

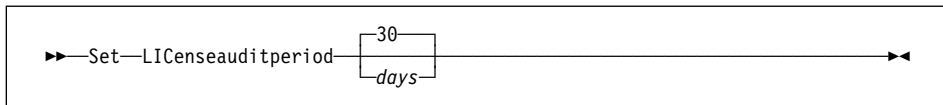
### SET LICENSEAUDITPERIOD (Set License Audit Period)

Use the SET LICENSEAUDITPERIOD command to specify the period, in days, between automatic license audits performed by the ADSM server.

#### Privilege Class

To issue this command, you must have system privilege.

#### Syntax



#### Parameters

*days*

Specifies the number of days between automatic server license audits on the ADSM server. This parameter is optional. The default value is 30. You can specify an integer in the range of 1 to 30, inclusive.

#### Examples

##### Task

Specify the license audit period for every 14 days.

**Command:** `set licenseauditperiod 14`

##### Output:

```
ANR2814I SET LICENSEAUDITPERIOD: License Audit Period
changed to 14 days.
```

## SET LICENSEAUDITPERIOD

### Related Commands

Table 145 provides references to related commands.

*Table 145. Commands Related to SET LICENSEAUDITPERIOD*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LICENSES	Checks for compliance with defined licenses.	Page 54
QUERY AUDITOCCUPANCY	Displays the data storage utilization for a client node.	Page 325
QUERY LICENSE	Displays information about licenses and audits.	Page 386
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
REGISTER LICENSE	Registers a new license with the ADSM server.	Page 463

## SET LOGMODE

---

### SET LOGMODE (Set Mode for Saving Recovery Log Records)

Use the SET LOGMODE command to set the mode for saving recovery log records. The log mode you specify determines how long ADSM saves records in the recovery log, which determines what kind of database recovery you can use. If the log mode is set to NORMAL, you can only restore a database to a specific point in time (point-in-time recovery). If the log mode is set to ROLLFORWARD, you can choose to restore a database to a specific point in time or to its most current state through the use of recovery log records (roll-forward recovery).

If the log mode is set to NORMAL, ADSM saves only those records needed to restore the database to the point of the last backup and resume database processing, if a system failure occurs. ADSM deletes any unnecessary records from the recovery log. Changes made to the database since the last backup cannot be recovered. Any backup versions of the database created by issuing the BACKUP DB command can only be used to perform point-in-time recovery.

If the log mode is set to ROLLFORWARD, ADSM saves all recovery log records that contain changes made to the database since the last time it was backed up. ADSM deletes recovery log records only when they become unnecessary because the database is successfully backed up. The recovery log records can be used to restore a database to its most current state after loading the most current database backup series.

A database backup series created while the log mode is set to ROLLFORWARD can be used to perform point-in-time recovery or roll-forward recovery.

If you specify ROLLFORWARD for the log mode, you can use the DEFINE DBBACKUPTRIGGER command to specify how much space can be used in the recovery log before ADSM automatically runs a backup of the database and removes any unnecessary records from the recovery log. Setting a database backup trigger helps to ensure that the recovery log does not run out of space for new records before the next backup is run.

For more information on how the log mode affects database recovery, see *ADSM Administrator's Guide*.

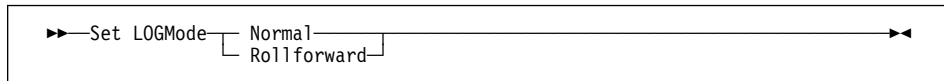
You cannot use the SET LOGMODE command while the BACKUP DB command is running. You must wait for the backup to complete or cancel the background process started by the BACKUP DB command before you can change the log mode.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

## SET LOGMODE

### Syntax



### Parameters

#### Normal

Specifies that ADSM saves only those recovery log records that are needed to recover the database in case of a system failure. This is the default for the log mode.

When the log mode is NORMAL, no automatic backups are performed based on the database backup trigger. ADSM deletes recovery log records that were required to perform roll-forward recovery or to restore a single database volume. However, you can still use existing database backups to perform point-in-time recovery. ADSM keeps log records necessary to resume database processing if a system failure occurs.

In response to the change to NORMAL mode, ADSM may issue a message warning that only point-in-time database recovery will be possible. If the database backup trigger is set, the message also warns that the trigger will be disabled. If issued from an administrative client (not from the server console), ADSM asks whether to proceed with the change to NORMAL mode.

#### Rollforward

Specifies that ADSM saves enough log records to ensure that the database can be restored to its most current state using the most recent database backup series and recovery log records. If issued from an administrative client (not from the server console), ADSM asks whether to proceed with the change to ROLLFORWARD mode.

If the database trigger is defined, ADSM automatically and immediately runs a full backup.

If the database trigger is not defined when you change the log mode from NORMAL to ROLLFORWARD, a full backup is required the next time a backup is run. If you do not specify TYPE=FULL the next time you issue the BACKUP DB command, the command fails.

### Examples

#### Task 1

Change the log mode from roll-forward to normal after the database backup trigger has been defined.

**Command:** set logmode normal

## SET LOGMODE

### Output:

```
ANR2294I Log mode set to NORMAL.
```

### Task 2

Change the log mode from normal to roll-forward. No database backup trigger is defined.

**Command:** set logmode rollforward

### Output:

```
ANR2294I Log mode set to ROLLFORWARD.
```

## Related Commands

Table 146 provides references to related commands.

*Table 146. Commands Related to SET LOGMODE*

Command	Description	Reference
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
UPDATE DBBACKUPTRIGGER	Changes one or more attributes of the database backup trigger.	Page 547



## SET MAXCMDRETRIES

---

### SET MAXCMDRETRIES (Set Number of Times Scheduler Retries Commands)

Use the SET MAXCMDRETRIES command to specify the maximum number of times that a client scheduler retries after a failed attempt to execute a scheduled command.

The MAXCMDRETRIES parameter only restricts the number of attempts that are made to retry a scheduled command that fails. This parameter does not limit the number of attempts made by the client scheduler to contact the server to obtain a schedule.

The MAXCMDRETRIES parameter can be specified by each user at the time their client scheduler program is started. Refer to the appropriate client documentation for more information on setting the maximum command retries value from the client.

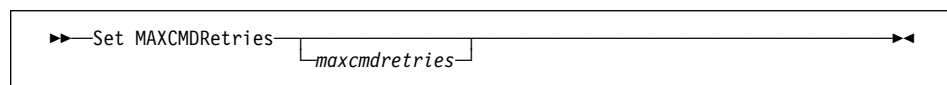
You can use the SET MAXCMDRETRIES command to set a global value for the maximum number of retries, which overrides the value specified by the user. The client's value is overridden only if the client can contact the server.

You can display the current value for the maximum command retries with the QUERY STATUS command. At installation, ADSM is configured so that each client determines its own value for this setting.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *maxcmdretries*

Specifies the maximum number of times the client scheduler retries commands after a failed attempt to execute a scheduled command. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive. If you do not specify a value for this parameter, each client determines its own value for this parameter.

### Examples

#### Task 1

Have all client schedulers retry a failed attempt to execute a scheduled command only twice.

**Command:** `set maxcmdretries 2`

## SET MAXCMDRETRIES

### Output:

```
ANR2525I Maximum number of command retries set to 2.
```

### Task 2

Have each client determine how often to retry a failed command.

**Command:** set maxcmdretries

### Output:

```
ANR2526I Maximum number of command retries reset to value  
determined by each client.
```

## Related Commands

Table 147 provides a reference to a related command.

*Table 147. Command Related to SET MAXCMDRETRIES*

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## SET MAXSCHEDESESSIONS

---

### SET MAXSCHEDESESSIONS (Set Maximum Scheduled Sessions)

Use the SET MAXSCHEDESESSIONS command to regulate the number of sessions that the server can use for processing scheduled work. This command specifies the maximum number of scheduled sessions as a percentage of the total number of server sessions available.

This setting should be changed if client nodes receive messages that server sessions are not available when trying to execute scheduled events. If you find that scheduled sessions are not contacting the server, you may have a network error.

If you find that the number of sessions for scheduled operations is insufficient, you can increase either the total number of sessions or the maximum number of scheduled sessions. However, increasing the total number of sessions available can adversely affect server performance, and increasing the maximum number of scheduled sessions can reduce the server's opportunity to execute nonscheduled work.

You can use the QUERY STATUS command to display the current value for the maximum number of sessions and the maximum number of scheduled sessions. At installation, the value for the maximum number of scheduled sessions is set to 50 percent.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►—Set MAXSCHedsessions—schedulepercent—◄
```

### Parameters

#### *schedulepercent*

Specifies an integer that gives the percentage of total server sessions that can be used for the processing of scheduled work. This parameter is required. You can specify an integer in the range of 0 to 100, inclusive. The maximum number of total available server sessions is determined by the MAXSESSIONS parameter in the server's option file.

If you set the maximum number of scheduled sessions to a value of 0, no scheduled events can begin. If you set the maximum number of scheduled sessions to a value of 100, the MAXSESSIONS parameter determines the maximum number of scheduled sessions.

## SET MAXSCHEDESESSIONS

### Examples

#### Task

By setting MAXSESSIONS to 80, you limit the server to 80 active sessions at one time. You want only 25 percent of your total available sessions to be available for scheduled activity. The maximum number of client/server sessions available for scheduled work at any given time is 20 sessions.

**Command:** `set maxschedsessions 25`

#### Output:

```
ANR2520I Scheduled sessions set to 25 percent.
```

### Related Commands

Table 148 provides references to related commands.

*Table 148. Commands Related to SET MAXSCHEDESESSIONS*

Command	Description	Reference
QUERY OPTION	Displays information about one or more options from the server options file.	Page 412
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## SET PASSEXP

---

### SET PASSEXP (Set Password Expiration Date)

Use the SET PASSEXP command to set the password expiration period. The password expiration period applies to both administrator and client node passwords.

Use the QUERY STATUS command to display the current password expiration period. At installation, this value is set to 90 days.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
▶—Set PASSExp—days—▶
```

### Parameters

*days*

Specifies an integer that defines the number of days for which the password remains valid. This parameter is required. You can specify an integer in the range of 1 to 9999, inclusive. If an administrator or client node password expires, the server prompts for a new password when the administrator or client node attempts to contact the server.

### Examples

## SET PASSEXP

### Task

Set the administrator and client node password expiration period to 45 days.

**Command:** `set passexp 45`

### Output:

```
ANR2092I Password expiration period set to 45 days.
```

## SET PASSEXP

### Related Commands

Table 149 provides references to related commands.

*Table 149. Commands Related to SET PASSEXP*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
UPDATE ADMIN	Changes the password or contact information associated with any administrator.	Page 537

## SET QUERYSCHEDPERIOD

---

### SET QUERYSCHEDPERIOD (Set Query Period for Polling Client Nodes)

Use the SET QUERYSCHEDPERIOD command to regulate the frequency with which client nodes contact the server to obtain scheduled work when they are running in the client-polling mode.

The value for the QUERYSCHEDPERIOD parameter can be set by each client node at the time the client scheduler program is started.

You can set a global value for the period between attempts by the client to contact the server for scheduled work. This value overrides the value specified by the client.

The client's value is only overridden if the client can contact the server.

You can set this period to correspond to the frequency with which the schedule or client association changes are being made. If client nodes poll more frequently for schedules, changes to scheduling information (through administrator commands) are propagated more quickly to client nodes. However, increased polling by the client nodes also increases network traffic.

You can display the value for the period between schedule queries with the QUERY STATUS command. At installation, ADSM is configured so that each client node determines its own value for this setting.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►► Set QUERYSChedperiod queryschedperiod ◀◀
```

### Parameters

*queryschedperiod*

Specifies the maximum number of hours between attempts by the client to contact the server for scheduled commands. This parameter is optional. You can specify an integer in the range of 1 to 9999, inclusive. If you do not specify a value for this parameter, each client determines its own value for this parameter.

### Examples

#### Task 1

Have all clients using the POLLING scheduling mode contact the server every 24 hours.

**Command:** set queryschedperiod 24

**Output:**



## SET QUERYSCHEDPERIOD

```
ANR2523I Schedule query period set to 24 hour(s).
```

### Task 2

Have all clients using the POLLING scheduling mode set their own query scheduling period.

**Command:** set queryschedperiod

**Output:**

```
ANR2524I Schedule query period reset to value determined by client.
```

### Related Commands

Table 150 provides references to related commands.

*Table 150. Commands Related to SET QUERYSCHEDPERIOD*

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
SET SCHEDMODES	Specifies the central scheduling mode for the server.	Page 532

## SET RANDOMIZE

---

### SET RANDOMIZE (Randomly Distribute Scheduled Start Times)

Use the SET RANDOMIZE command to specify the degree to which start times are randomized within the startup window of each schedule for clients using the client-polling mode.

Each schedule has an associated window during which execution can begin. To balance network and server load, the start times for individual clients can be scattered across that window. You can use this command to specify the fraction of the window over which start times for individual clients are to be distributed. The randomization occurs at the beginning of the window to allow time for retries, if necessary, unless the client's first contact with the server is after the start time for the event.

Set the randomization percentage to a value greater than 0 to prevent communication errors that can result from a large group of clients contacting the server simultaneously. If you do experience communication errors, you can increase the randomization percentage so that client contact is spread out, thus decreasing the chance for communication overload and failure.

It is possible, especially after a client or the server has been restarted, that a client may not poll the server until *after* the beginning of the time window in which the next scheduled event is to start. In this case, the starting time is randomized over the specified percentage of the *remaining* duration of the start window.

For example, consider the following situation:

- Randomization is set at 50 percent
- The startup window for a particular event is from 8:00 to 9:00
- Ten clients are associated with the schedule
- Nine clients poll the server before 8:00
- One client does not poll the server until 8:30

The nine clients that polled the server *before* the beginning of the startup window are assigned randomly selected starting times between 8:00 and 8:30. The client that polled at 8:30 receives randomly selected starting time that is between 8:30 and 8:45.

You can display the value for the schedule randomization percentage with the QUERY STATUS command. At installation, this value is set to 25 percent.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set RANDomize—randomizepercent—————◄◄
```

## SET RANDOMIZE

### Parameters

#### *randomizepercent*

Indicates the percentage of the startup window over which the start times for individual clients are distributed. This value is specified as an integer in the range of 0 to 50, inclusive.

A value of 0 indicates that no randomization occurs and that all clients execute the schedule at the beginning of the window. A value of 50 indicates that clients are assigned start times that are randomly scattered across the first half of the startup window.

At installation, this value is set to 25, indicating that the first 25 percent of the window is used for randomization.

If you have specified DURUNITS=INDEFINITE in the DEFINE SCHEDULE command, the percentage specified by the RANDOMIZEPERCENT parameter is regarded as a fraction of a one-day time period. That is, if the value for the RANDOMIZEPERCENT parameter is 25, a schedule with an indefinite window duration has start times scattered over a 6-hour time range ( $0.25 \times 24$ ).

### Examples

#### Task

Set randomization to 50 percent.

**Command:** set randomize 50

#### Output:

```
ANR2522I Randomization set to 50 percent.
```

### Related Commands

Table 151 provides references to related commands.

Table 151. Commands Related to SET RANDOMIZE

Command	Description	Reference
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
SET SCHEDMODES	Specifies the central scheduling mode for the server.	Page 532

## SET REGISTRATION

---

### SET REGISTRATION (Set Open or Closed Registration)

Use the SET REGISTRATION command to either require a system or policy administrator to register client nodes (closed registration), or permit each user to register his or her own workstation as a client node with the server (open registration).

With *closed* registration, a system or policy administrator defines:

- Each workstation's node name
- Each user's password
- The policy domain to which the client node belongs
- Whether the user can choose to compress files before sending them to server storage
- Whether the user is allowed to delete backup or archive files from server storage

With *open* registration, when a user attempts to access the server from an unregistered client node, the server prompts the user for a node name, password, and contact information, and registers the workstation. The server sets the following defaults:

- Each client node is assigned to the policy domain named STANDARD.
- Each user defines whether data compression is used before files are sent to server storage.
- Each user is allowed to delete archived files from server storage. The user cannot delete backup files.
- The administrator can reassign domains or change node attributes using the UPDATE NODE command.

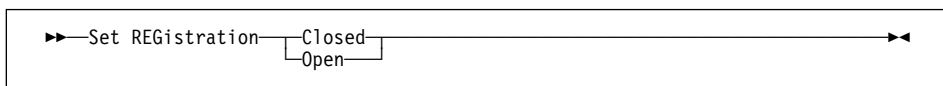
Existing registered client nodes are not affected by changes in the registration process.

Use the QUERY STATUS command to display the status of registration. At installation, this value is set to CLOSED.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node belongs.

### Syntax



### Parameters

#### Closed

Specifies that client nodes must be registered through a system or policy administrator.

## SET REGISTRATION

### Open

Specifies that users can register their workstations as client nodes with the server.

## Examples

### Task 1

Have a policy or system administrator register all client nodes to the server.

**Command:** set registration closed

**Output:**

```
ANR2093I Registration parameter set to CLOSED.
```

### Task 2

Have all users register themselves as client nodes to the server.

**Command:** set registration open

**Output:**

```
ANR2093I Registration parameter set to OPEN.
```

## Related Commands

Table 152 provides a reference to a related command.

*Table 152. Command Related to SET REGISTRATION*

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## SET RETRYPERIOD

---

### SET RETRYPERIOD (Set Time between Retry Attempts)

Use the SET RETRYPERIOD command to specify the number of minutes between attempts by the client scheduler to retry after a failed attempt to contact the server or to execute a scheduled command.

Each client can set their own retry period at the time their client scheduler program is started. You can use this command to set a global value for the retry period which will override the value specified by all clients. The client's value is overridden only if the client is able to connect with the server.

When setting the period between retry attempts, set a time period that permits more than one retry attempt within a typical startup window.

This command is used in conjunction with the SET MAXCMDRETRIES command to regulate the period of time and the number of retry attempts to execute a failed command.

You can display the value for the period between retries by issuing the QUERY STATUS command. At installation, ADSM is configured so that each client determines its own value for this setting.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►► Set RETRYPeriod [retryperiod] ◄◄
```

### Parameters

#### *retryperiod*

Specifies the number of minutes between attempts by the client scheduler to retry after a failed attempt to contact the server or to execute a scheduled command. You can specify an integer in the range of 1 to 9999, inclusive. If you do not specify a value for this parameter, each client determines its own value for this parameter.

### Examples

#### Task 1

Allow each client to specify their own period between retries.

**Command:** set retryperiod

## SET RETRYPERIOD

### Output:

```
ANR2528I Retry period set to value determined by each client.
```

### Task 2

Have the client scheduler retry failed attempts to contact the server or to execute scheduled commands every 15 minutes.

**Command:** `set retryperiod 15`

### Output:

```
ANR2527I Retry period set to 15 minute(s).
```

## Related Commands

Table 153 provides references to related commands.

*Table 153. Commands Related to SET RETRYPERIOD*

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
SET MAXCMDRETRIES	Specifies the maximum number of retries after a failed attempt to execute a scheduled command.	Page 517

## SET SCHEDMODES

---

### SET SCHEDMODES (Select a Central Scheduling Mode)

Use the SET SCHEDMODES command to select the central scheduling modes that the server allows. Use this command when you set up central scheduling to determine how the clients communicate with the server to begin scheduled work.

You can restrict clients from using a scheduling mode with this command.

With client-polling scheduling, a client queries the server at prescribed time intervals to obtain scheduled work or to ensure that the schedules they are waiting to execute have not changed. At the scheduled start time, the client performs the scheduled backup or archive operations, sending the results to the server.

With server-prompted scheduling, clients wait to be contacted by the server when scheduled work needs to be performed and a session is available. When contacted, the client performs the scheduled backup or archive operations, sending the results to the server. The server-prompted mode can only be used between clients and servers that communicate through TCP/IP.

To select both server-prompted and client-polling modes, specify ANY.

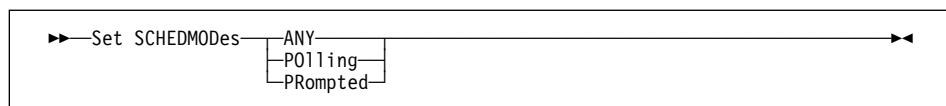
It is very important that you communicate to all users the scheduling modes allowed by the server.

You can display the value for the scheduling mode supported by issuing the QUERY STATUS command. At installation, this value is set to ANY.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### ANY

Specifies that the server allows clients to run in either the client-polling or the server-prompted scheduling mode. Each client must be configured to select the scheduling mode in which it operates.

#### POLLing

Specifies that only the client-polling mode can be used. Client nodes poll the server at prescribed time intervals to obtain scheduled work. If this mode is selected, configure each client to use the client-polling mode.



## SET SCHEDMODES

### PRompted

Specifies that only the server-prompted mode can be used. This mode is only available for clients that communicate with TCP/IP. Client nodes wait to be contacted by the server when scheduled work needs to be performed and a session is available. If this mode is selected, configure each client to use the server-prompted mode.

## Examples

### Task 1

A server allows both server-prompted and client-polling central scheduling. You want to temporarily restrict the scheduled operations to clients that use the client-polling mode. If you set the schedule mode to POLLING, the server discontinues prompting clients to execute scheduled commands. This means that any client scheduler using the server-prompted mode waits until you set the schedule mode to ANY or PROMPTED.

**Command:** `set schedmodes polling`

#### Output:

```
ANR2529I Scheduling mode set to POLLING.
```

### Task 2

Set the central scheduling mode to PROMPTED. Only clients that communicate with TCP/IP and that are configured to use the server-prompted mode will perform scheduled operations.

**Command:** `set schedmodes prompted`

#### Output:

```
ANR2530I Scheduling mode set to PROMPTED.
```

## Related Commands

Table 154 provides a reference to a related command.

Table 154. Command Related to SET SCHEDMODES

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

## SET SERVERNAME

---

### SET SERVERNAME (Specify the Server Name)

Use the SET SERVERNAME command to specify the name by which the server is to be identified. At installation, the name of the server is set to ADSM. After installation, this command can be used to modify the name of the server.

You can use the QUERY STATUS command to see the name of the server.

The server name is reported to a client node when it contacts the server.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set SERVERname—servername—◄◄
```

### Parameters

*servername*

Specifies the name by which the server is to be identified. This parameter is required. The maximum length of the name is 64 characters.

### Examples

#### Task

Name the server WELLS\_DESIGN\_DEPT.

**Command:** set servername wells\_design\_dept

#### Output:

```
ANR2094I Server name set to WELLS_DESIGN_DEPT.
```

### Related Commands

Table 155 provides a reference to a related command.

---

*Table 155. Command Related to SET SERVERNAME*

---

Command	Description	Reference
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431

---

## UNLOCK ADMIN

---

### UNLOCK ADMIN (Unlock an Administrator)

Use the UNLOCK ADMIN command to allow a locked administrator to reaccess the server.

By using the LOCK ADMIN and UNLOCK ADMIN commands, you can temporarily restrict an administrator from accessing the server.

### Privilege Class

To issue this command, you must have system privilege.

### Syntax

```
▶—UNLOCK Admin—adminname—▶
```

### Parameters

*adminname*

Specifies the name of the administrator to be unlocked. This parameter is required.

### Examples

#### Task

The administrator JOE is locked out of ADSM. Permit JOE to reaccess the server.

**Command:** unlock admin joe

#### Output:

```
ANR2074I Administrator JOE unlocked.
```

### Related Commands

Table 156 provides references to related commands.

Table 156. Commands Related to UNLOCK ADMIN

Command	Description	Reference
LOCK ADMIN	Prevents an administrator from accessing ADSM.	Page 303
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317

## UNLOCK NODE

---

### UNLOCK NODE (Unlock a Client Node)

Use the UNLOCK NODE command to allow a locked client node to reaccess the server.

By using the LOCK NODE and UNLOCK NODE commands, you can temporarily restrict a client node from accessing the server.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node is assigned.

### Syntax

```
▶▶—UNLOCK Node—nodename————▶▶
```

### Parameters

*nodename*

Specifies the name of the client node to be unlocked. This parameter is required.

### Examples

#### Task

The client node SMITH is locked out of ADSM. Permit SMITH to reaccess the server.

**Command:** unlock node smith

#### Output:

```
ANR2066I Node SMITH unlocked.
```

### Related Commands

Table 157 provides references to related commands.

---

*Table 157. Commands Related to UNLOCK NODE*

Command	Description	Reference
LOCK NODE	Prevents a client from accessing the server.	Page 304
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403

---

## UPDATE ADMIN

---

### UPDATE ADMIN (Update an Administrator)

Use the UPDATE ADMIN command to change the password or contact information for an administrator. However, you cannot update the server console administrator name.

You can use this command to modify selected parameters for the specified administrator. If you do not explicitly update a parameter, it remains unchanged.

You must use the RENAME ADMIN command to change the name of a registered administrator.

Passwords for both administrative and client node sessions must be changed after a length of time determined by the password expiration period. The SET PASSEXP command is used to set this period.

### Privilege Class

To issue this command to change another administrator's password or contact information, you must have system privilege. Any administrator can issue this command to update his or her own password or contact information.

### Syntax

► UPDATE ADMIN <sup>(1)</sup> *adminname* [*password*] [CONTACT=*contactinfo*] ◀◀

**Note:**  
1 You must specify at least one optional parameter on this command.

### Parameters

#### *adminname*

Specifies the name of the administrator to be updated. This parameter is required.

#### *password*

Specifies the administrator's password. This parameter is optional. Passwords remain current for a period determined by the password expiration period.

#### **CONTACT**=*contactinfo*

Specifies a text string that identifies the administrator. This parameter is optional. The text string must be enclosed in quotation marks if it contains any blanks. To remove previously defined contact information, specify a null string ("") for this value.

## UPDATE ADMIN

### Examples

#### Task

Update the administrator LARRY to have the password SECRETWORD.

**Command:** update admin larry secretword

#### Output:

```
ANR2071I Administrator LARRY updated.
```

### Related Commands

Table 158 provides references to related commands.

*Table 158. Commands Related to UPDATE ADMIN*

Command	Description	Reference
QUERY ADMIN	Displays information about one or more ADSM administrators.	Page 317
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.	Page 431
RENAME ADMIN	Changes an ADSM administrator's name.	Page 474
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.	Page 521

## UPDATE COPYGROUP

### UPDATE COPYGROUP (Update a Copy Group)

Use the UPDATE COPYGROUP command to update a defined backup or archive copy group.

You can use this command to modify selected parameters for the specified copy group. If you do not explicitly update a parameter, it remains unchanged.

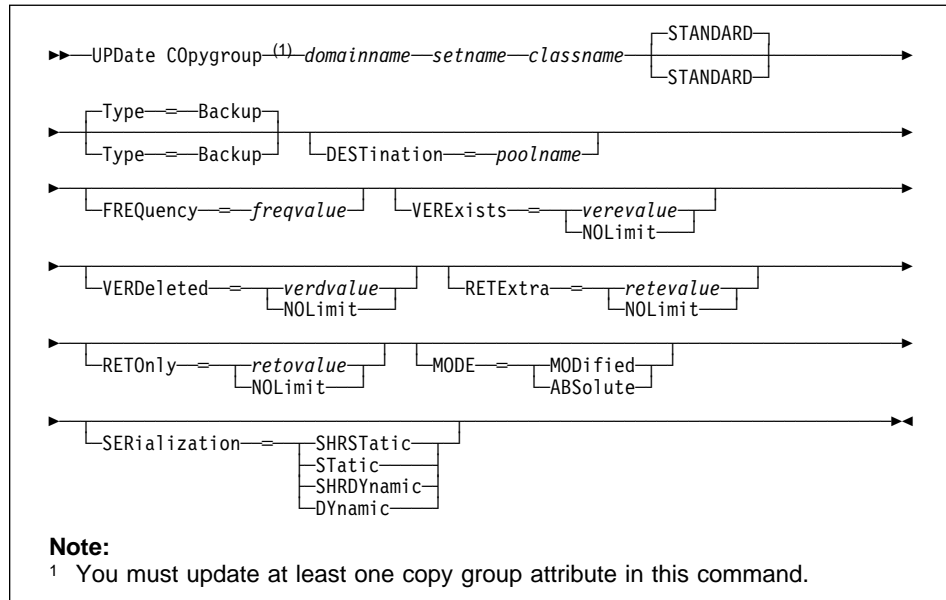
The UPDATE COPYGROUP command will fail if a copy storage pool name is specified as a destination.

The UPDATE COPYGROUP command takes two forms, depending upon whether the update is for a backup copy group or for an archive copy group. The syntax for each form is defined separately below.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the copy group belongs.

### Syntax for Updating a Backup Copy Group



### Parameters for Updating a Backup Copy Group

*domainname*

Specifies the name of the policy domain in which the copy group is defined. This parameter is required.

## UPDATE COPYGROUP

### *setname*

Specifies the name of the policy set in which the copy group to be updated is defined. This parameter is required.

### *classname*

Specifies the name of the management class to which the copy group belongs. This parameter is required.

### **STANDARD**

Specifies the name of the copy group. The name of the copy group must be STANDARD. The default value is STANDARD.

### **Type=Backup**

Specifies that you want to update a backup copy group. This parameter is optional. The default value is BACKUP.

### **DESTination=poolname**

Specifies the name of the primary storage pool in which backup data is to be stored. This parameter is optional. You can only specify one storage pool as the destination for a copy group.

### **FREQUENCY=freqvalue**

Specifies the minimum number of days that must elapse between successive backups. This parameter is optional. ADSM does not perform a backup unless the specified number of days has elapsed since the last backup. You can specify a value from 0–9999, inclusive.

Backup frequency works in conjunction with the MODE parameter, which is described later.

### **VERExists=vervalue**

Specifies the maximum number of different backup versions to retain for files that currently exist on the client file system. This parameter is optional.

If the maximum number of versions allowed is 5, and backup processing creates a sixth version, ADSM deletes the oldest backup version from server storage. Possible values are:

#### *vervalue*

Specifies the number of backup versions to retain for files that currently exist on the client file system. You can specify a value from 1–9999, inclusive.

### **NOLimit**

Specifies that you want ADSM to retain all backup versions.

### **VERDeleted=verdvalue**

Specifies the maximum number of backup versions to retain for files that have been deleted from the client file system after being backed up to the server. This parameter is optional. ADSM ignores this parameter if the file still resides on the client node.

If the client file is deleted, during the next backup ADSM changes the active backup version to inactive and erases the oldest versions that are in excess of



## UPDATE COPYGROUP

the number specified by this parameter. The expiration date for the remaining versions is determined by the values specified for the RETEXTRA and RETONLY parameters. Possible values are:

### *vervalue*

Specifies the number of backup versions to retain for files that have been deleted from the client file system after being backed up to the server. You can specify a value from 0–9999, inclusive.

### **NOLimit**

Specifies that you want ADSM to retain all backup versions for files that are deleted from the client file system after being backed up to the server.

### **RETE***Extra=retevalue*

Specifies the retention time, in days, that ADSM retains all but the most recent backup version for files that exist on the client workstation. The most recent version is maintained indefinitely, or until the client file is deleted. This parameter is optional. Possible values are:

### *retevalue*

Specifies the number of days to retain all backup versions other than the active version. You can specify a value from 0–9999, inclusive. The default value is 30 days.

### **NOLimit**

Specifies that you want ADSM to retain all backup versions, other than the most recent version, indefinitely (that is, inventory expiration processing does not affect the backup versions). The most recent version of a file that exists on the client workstation is retained indefinitely or until the user deletes the file on the workstation. If the user deletes the file on the workstation, the file is rolled off using the value specified for the VERDELETED parameter.

If the NOLIMIT value is specified, extra backup versions are deleted during backup processing under control of the VEREXISTS or VERDELETED parameters.

### **RETO***nly=retovalue*

Specifies the retention time, in days, that ADSM retains the last copy it has of a file that has been deleted from the client file system. This parameter is optional. Possible values are:

### *retovalue*

Specifies the number of days to retain the last remaining inactive copy of a file. You can specify a value from 0–9999, inclusive.

### **NOLimit**

Specifies that the last remaining inactive copy of a file be kept indefinitely.

If you specify NOLIMIT, only backup versions are deleted under control of the VEREXISTS or VERDELETED parameters.

## UPDATE COPYGROUP

### **MODE**=*mode*

Specifies whether a file should be backed up based on changes made to the file since the last time it was backed up or regardless of whether it has changed. This parameter is only used by incremental backup. This parameter is optional. Possible values are:

#### **MODified**

Specifies that the file is backed up only if it has changed since the last backup. A file is considered changed if any of the following is true:

- The date last modified is different
- The file size is different
- The file owner is different
- The file permissions are different

#### **ABSolute**

Specifies that the file is backed up regardless of whether it has been changed.

### **SERialization**=*serialvalue*

Specifies whether a file or directory can be in use during a backup operation and what the server should do if it is in use. This parameter is optional. The default value is SHRSTATIC.

**Attention:** Be careful about choosing a management class that contains a copy group that specifies SERIALIZATION=SHRDYNAMIC or SERIALIZATION=DYNAMIC.

If you choose a management class that allows a file to be backed up while it is in use, the backup version stored on the server might be a fuzzy backup. A fuzzy backup is a backup version that may not accurately reflect what is currently in the file. It might contain some, but not all, changes. If that is not acceptable, be sure to choose a management class that allows ADSM to create a backup version only if the file is not being used.

If a file is restored that contains a fuzzy backup, the file might not be usable, depending on the file's application.

Possible values are:

#### **SHRStatic**

Specifies that a file or directory must not be backed up while being modified. The server attempts to perform a backup as many as four times, depending on the value specified for CHANGINGRETRIES specified in the client user options file. If the file or directory is in use during the backup process, the server will not back it up.

#### **Static**

Specifies that a file or directory must not be modified during backup. If the file or directory changes during the backup attempt, it is not backed up.

Platforms that do not support the STATIC option default to SHRSTATIC. The STATIC option is not supported on the DOS platform.

## UPDATE COPYGROUP

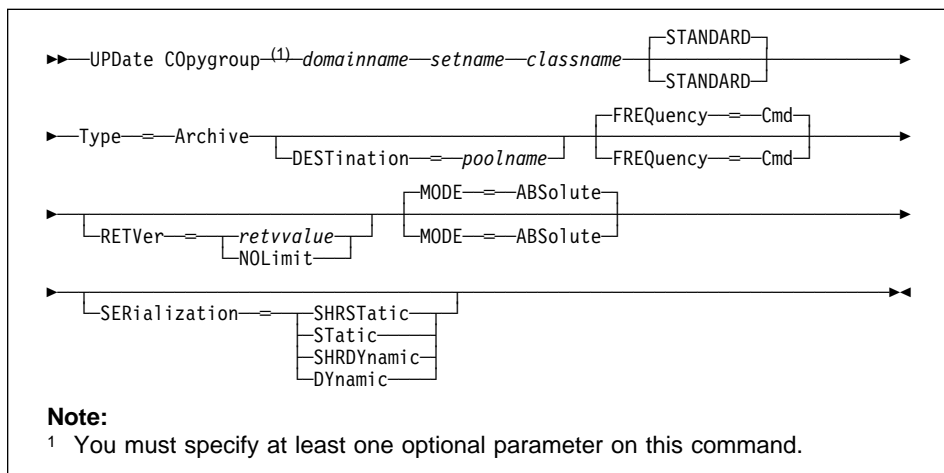
### SHRDYnamic

Specifies that if the file or directory is being modified during a backup attempt, the server will back it up on its last try even though the file or directory is in use.

### DYnamic

Specifies that a file or directory is backed up on the first attempt by the server, regardless if the file or directory is in use during the backup process.

### Syntax for Updating an Archive Copy Group



### Parameters for Updating an Archive Copy Group

#### *domainname*

Specifies the name of the policy domain in which the copy group is defined. This parameter is required. The maximum length of this name is 30 characters.

#### *setname*

Specifies the name of the policy set in which the copy group belongs. This parameter is required. The maximum length of this name is 30 characters.

#### *classname*

Specifies the name of the management class to which the copy group belongs. This parameter is required. The maximum length of this name is 30 characters.

#### **STANDARD**

Specifies the name of the copy group. The name of the copy group must be STANDARD. The default value is STANDARD.

#### **Type=Archive**

Specifies that you want to update an archive copy group. This parameter is required.

## UPDATE COPYGROUP

### **DESTination**=*poolname*

Specifies the name of the storage pool where the archive files are to reside. This parameter is optional. You can only specify one storage pool as the destination for a copy group.

### **FREQuency**=**Cmd**

Specifies that a file archived. The copy frequency must be CMD. This parameter is optional. A file is archived only when specified in an ARCHIVE command or when selected for archive using the graphical user interface.

### **RETVer**=*retvvalue*

Specifies the number of days to retain an archive copy that is bound to this management class. The value of this parameter determines whether a copy is deleted during inventory expiration processing. This parameter is optional.

#### *retvvalue*

Specifies the number of days to retain an archive copy. You can specify a value from 0–9999, inclusive.

### **NOLimit**

Specifies that an archive copy is maintained indefinitely (that is, inventory expiration processing does not affect the archive copy).

### **MODE**=*ABSolute*

Specifies that the file is archived regardless of whether it has been changed since the last archive request. This parameter is optional. The default value is ABSOLUTE.

### **SERialization**=*serialvalue*

Specifies whether a file can be in use during an archive operation and what the server should do if it is in use. This parameter is optional. The default value is SHRSTATIC.

**Attention:** Be careful about choosing a management class that contains a copy group that specifies **SERIALIZATION=SHRDYNAMIC** or **SERIALIZATION=DYNAMIC**.

If you choose a management class that allows a file to be archived while it is in use, the archive copy stored on the server might be a fuzzy copy. A fuzzy copy is an archive copy that may not accurately reflect what is currently in the file. It might contain some, but not all, changes. If that is not acceptable, be sure to choose a management class that allows ADSM to create an archive copy only if the file is not being used.

If a file is restored that contains a fuzzy copy, the file might not be usable, depending on the file's application.

Possible values are:

### **SHRStatic**

Specifies that a file must not be archived while being modified. The server attempts to perform an archive as many as four times, depending on the value specified for **CHANGINGRETRIES** specified in the client user options

## UPDATE COPYGROUP

file. If the file is in use during the archive process, the server will not archive.

### Static

Specifies that a file must not be modified during archive. If the file changes during the archive attempt, it is not archived.

Platforms that do not support the STATIC option default to SHRSTATIC. The STATIC option is not supported on the DOS platform.

### SHRDynamic

Specifies that if the file is being modified during an archive attempt, the server will archive it on its last try even though the file is in use.

### Dynamic

Specifies that a file is archived on the first attempt by the server, regardless if the file is in use during the archive process.

## Examples

### Task 1

Update an archive copy group named STANDARD in the EMPLOYEE\_RECORDS policy domain, VACATION policy set, ACTIVEFILES management class. The archive destination is BACKUPPOOL where the archive copies are kept for 190 days.

**Command:** `update copygroup employee_records vacation activefiles standard type=archive destination=backuppoo1 retver=190`

#### Output:

```
ANR1537I Archive copy group STANDARD updated in policy domain
EMPLOYEE_RECORDS, set VACATION, management class ACTIVEFILES.
```

### Task 2

Update a backup copy group named STANDARD in the EMPLOYEE\_RECORDS policy domain, VACATION policy set, ACTIVEFILES management class. The backup destination is BACKUPPOOL, with a minimum interval of seven days between backups, regardless of whether the files have been modified. Retain up to two active backup versions.

**Command:** `update copygroup employee_records vacation activefiles standard type=backup destination=backuppoo1 frequency=7 verexists=2 mode=absolute`

#### Output:

```
ANR1532I Backup copy group STANDARD updated in policy domain
EMPLOYEE_RECORDS, set VACATION, management class ACTIVEFILES.
```

## UPDATE COPYGROUP

### Related Commands

Table 159 provides references to related commands.

*Table 159. Commands Related to UPDATE COPYGROUP*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ASSIGN DEFMGMTCLASS	Assigns an existing management class as the default for a specified policy set.	Page 48
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DELETE COPYGROUP	Deletes a backup or archive copy group from a policy domain and policy set.	Page 199
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
EXPIRE INVENTORY	Manually starts inventory expiration processing.	Page 248
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397

## UPDATE DBBACKUPTRIGGER

### UPDATE DBBACKUPTRIGGER (Update the Database Backup Trigger)

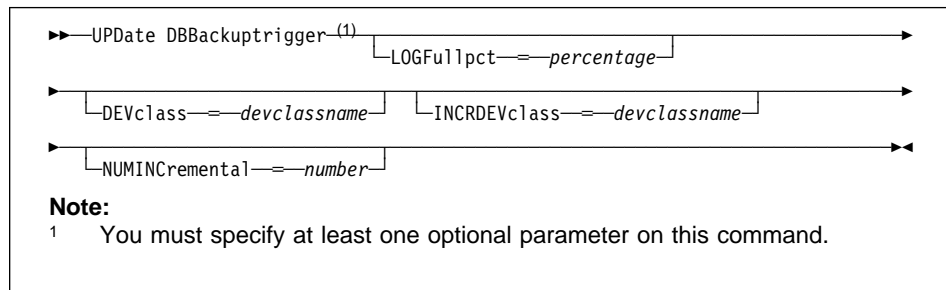
Use the UPDATE DBBACKUPTRIGGER command to change one or more parameters for the database backup trigger. You can use this command to change selected parameters. Any parameters you do not specify remain unchanged.

To display the current settings, use the QUERY DBBACKUPTRIGGER command.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax



### Parameters

#### LOGFullpct=*percentage*

Specifies when ADSM automatically backs up the database, based on the percentage of available space used for the recovery log. When the percentage of space used for the recovery log exceeds this value, ADSM performs an automatic backup of the database and removes unnecessary records from the recovery log. This parameter is optional. You can specify a value from 1–99. The default value is 50 percent.

#### DEVclass=*devclassname*

Specifies the name of the sequential access device class to use for backups. This parameter is optional.

If the mount limit for the device class specified has already been reached (all drives are busy) when the backup runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the backup.

The server options file that is created during installation includes a DEVCONFIG option that specifies an external file named devcnfg.out. ADSM stores a backup copy of device class definitions in this file. If your ADSM database is damaged or lost and must be restored, the definitions created using the DEFINE DEVCLASS command will not be available and will need to be recreated manually.

## UPDATE DBBACKUPTRIGGER

The server options file also includes a VOLUMEHISTORY option that specifies an external file (named volhist.out) in which to store a backup copy of sequential volume history information. The volume history file is also used for database recovery.

### **INCRDEVclass**=*devclassname*

Specifies the name of the sequential access device class to use for incremental backups. This parameter is optional. If you do not specify a value for this option, all backups are run using the device class specified with the DEVCLASS parameter.

If the mount limit for the device class specified has already been reached (all drives are busy) when the backup runs, ADSM automatically cancels lower priority operations, such as reclamation, to make a mount point available for the backup.

The server options file that is created during installation includes a DEVCONFIG option that specifies an external file named devcnfg.out. ADSM stores a backup copy of device class definitions in this file. If your ADSM database is damaged or lost and must be restored, the definitions created using the DEFINE DEVCLASS command will not be available and will need to be recreated manually.

### **NUMINcremental**=*number*

Specifies the maximum number of incremental backups that can be run before a full backup is required. This parameter is optional. You can specify a value from 0–32. A 0 specifies that ADSM runs only full backups automatically. The default value is 6.

Each incremental backup, whether run automatically by ADSM or run by using the BACKUP DB command, is added to the count of incremental backups run.

Each full backup, whether run automatically by ADSM or run by using the BACKUP DB command, resets the count for incremental backups to 0.

## Examples

### Task

Reset the database backup trigger at 60 percent and the number of incremental backups that ADSM runs between full backups to 5.

**Command:** update dbbackuptrigger logfull=60 numincremental=5

### Output:

```
ANR2285I Database backup trigger updated, but disabled.
```



## UPDATE DBBACKUPTRIGGER

### Related Commands

Table 160 provides references to related commands.

*Table 160. Commands Related to UPDATE DBBACKUPTRIGGER*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
DEFINE DBBACKUPTRIGGER	Defines a trigger for automatic ADSM database backups.	Page 112
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DELETE DBBACKUPTRIGGER	Deletes the database backup trigger.	Page 202
QUERY DBBACKUPTRIGGER	Displays information about the database backup trigger.	Page 345
SET LOGMODE	Sets the mode for saving records in the recovery log.	Page 514

## UPDATE DEVCLASS

---

### UPDATE DEVCLASS (Change the Attributes of a Device Class)

Use the UPDATE DEVCLASS command to update a defined device class. You can use this command to modify selected parameters for the specified device class. If you do not explicitly update a parameter, it remains unchanged. You can use the DEFINE DEVCLASS command to create a new device class or the DELETE DEVCLASS command to eliminate existing device classes.

The DISK device class is predefined by ADSM and requires no user intervention. This also means that the DISK device class cannot be modified or eliminated with the UPDATE DEVCLASS or DELETE DEVCLASS commands.

The supported parameters for this command vary across device type. For more information on updating device classes, refer to *ADSM Administrator's Guide*.

The server options file that is created during installation includes a DEVCONFIG option that specifies a default device configuration file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. The files you specify with this option are automatically updated with the results of this command. The files are also automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

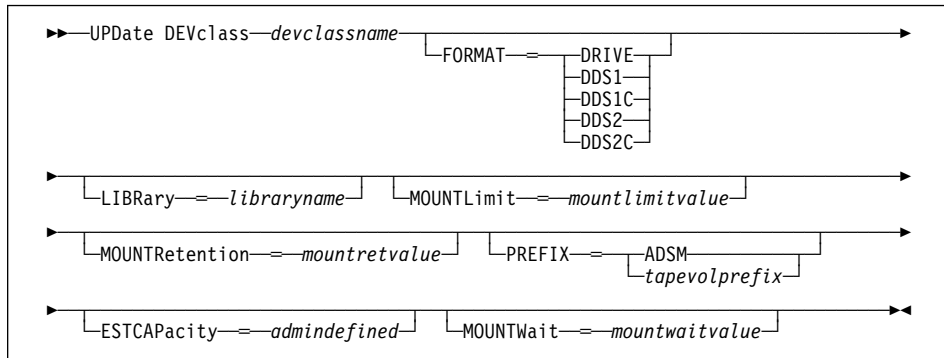
### Syntax

The supported parameters for this command vary between device types; therefore, the syntax and parameter descriptions are provided according to the device type. The syntax and parameter information is presented in the following order:

- 4MM
- 8MM
- DLT
- FILE
- LAN3995
- MO35
- OPTICAL
- QIC
- REMOVABLEFILE

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of 4MM



### Parameters for Updating a Device Class with a Device Type of 4MM

#### *devclassname*

Specifies the name of the device class to be defined. This parameter is required.

#### **FORMAT=tapeformat**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following are the possible values for 4mm devices:

#### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the `DRIVE` value when a mixture of devices are used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the `FORMAT=DRIVE` option.

#### **DDS1**

Specifies that ADSM writes data using the DDS-1 recording format for 4mm tapes, without compression. This format results in an uncompressed cartridge capacity of 1.3GB on 60 meter tape and 2GB on 90 meter tape. This format only applies to 60 and 90 meter tapes.

#### **DDS1C**

Specifies that ADSM writes data using the DDS-1 recording format for 4mm tapes, with compression enabled. This results in a cartridge capacity of approximately 1.3GB on a 60 meter tape, or 2GB on a 90 meter tape. This format, however, uses the tape drive hardware compression feature, so the actual capacity may be greater, depending on the effectiveness of compression. This format only applies to 60 and 90 meter tapes.

#### **DDS2**

Specifies that ADSM writes data using the DDS-2 recording format for 4mm tapes, without compression. This format results in a cartridge

## UPDATE DEVCLASS

capacity of 4GB on 120 meter tape. This format only applies to 120 meter tape.

### **DDS2C**

Specifies that ADSM writes data using the DDS-2 recording format for 4mm tapes, with compression. This format results in a cartridge capacity of approximately 8GB on 120 meter tape. This format, however, uses the tape drive hardware compression feature, so the actual capacity may be greater, depending on the effectiveness of compression. This format only applies to 120 meter tape.

### **LIBRARY=libraryname**

Specifies the name of the defined library object that contains the 4mm tape drives used by this device class. This parameter is optional. For information about defining a library object, see the DEFINE LIBRARY command.

### **MOUNTLimit=mountlimitvalue**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class.

If this value is set to 1, then it may not be possible for media-to-media operations (like reclamation) to work.

### **MOUNTRetention=mountretvalue**

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX=tapevolprefix**

Specifies the prefix of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The maximum length of this prefix is 8 characters.

If you have already established a media label naming convention that supports your current management system, consider using a volume prefix that conforms to your naming conventions.

### **ESTCapacity=admindefined**

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

## UPDATE DEVCLASS

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for 4mm tapes, see Table 161.

Table 161. ADSM Default Estimated Capacity for 4mm Tape

Tape Format	Estimated Capacity
DDS1	1.3 - 2.0GB
DDS1C	1.3 - 2.0GB (see note)
DDS2	4.0GB
DDS2C	8.0GB (see note)

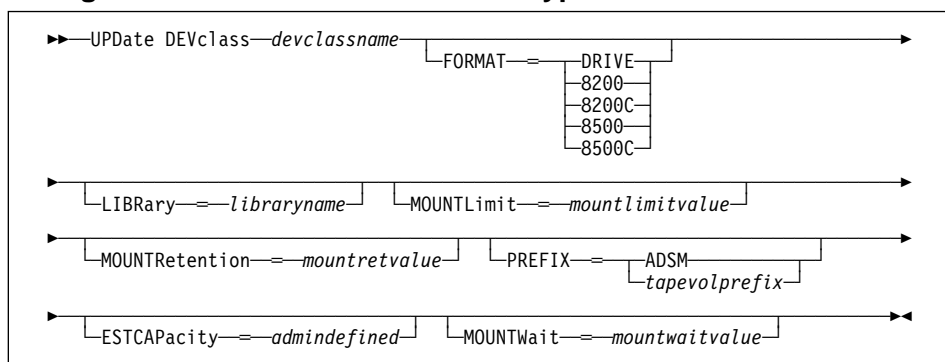
**Note:** This format uses the tape drive hardware compression feature, depending on the effectiveness of compression the actual capacity may be greater than the listed value.

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of 8MM



### Parameters

#### Parameters for Updating a Device Class with a Device Type of 8MM

##### *devclassname*

Specifies the name of the device class to be updated. This parameter is required.

##### **FORMAT=tapeformat**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

Possible values for 8MM devices are:

##### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices are used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

##### **8200**

Specifies that ADSM writes data using the 8200 recording format. This results in a tape capacity of 2.3GB when using standard 112M tape cartridges.

##### **8200C**

Specifies that ADSM writes data using the 8200C recording format. This results in a tape capacity of approximately 2.3GB when using standard 112 meter tape cartridges. This format, however, uses the tape drive's hardware compression feature, so the actual capacity may be greater than 2.3GB, depending on the effectiveness of the compression feature.

## UPDATE DEVCLASS

### **8500**

Specifies that ADSM writes data using the 8500 recording format. This results in a tape capacity of 5.0GB when using standard 112M tape cartridges.

### **8500C**

Specifies that ADSM writes data using the 8500C recording format. This results in a tape capacity of at least 5.0GB when using standard 112M tape cartridges. However, this format uses the tape drive's hardware compression feature, so the actual capacity of the cartridge may be greater than 5.0 GB, depending on the effectiveness of the compression feature.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the 8MM tape drives that can be used by this device class. For information about defining a library object see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class.

If this value is set to 1, then it may not be possible for media-to-media operations (like reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name which the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

If you have already established a media label naming convention that supports your current management system, consider using a volume prefix that conforms to your naming conventions.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You must specify this value as an integer followed by a scale factor. Possible scale factors are:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

## UPDATE DEVCLASS

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for 8MM tapes, see Table 162.

Table 162. ADSM Default Estimated Capacity for 8mm Tape

Tape Format	Estimated Capacity
8200	2.3GB
8200C	2.3GB (see note)
8500	5.0GB
8500C	5.0GB (see note)

**Note:** This format uses the tape drive hardware compression feature, depending on the effectiveness of compression the actual capacity may be greater than the listed value.

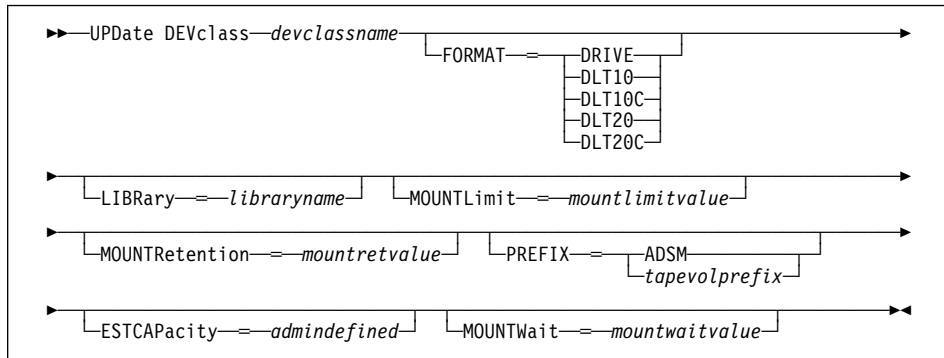
### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.



## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of DLT



### Parameters

#### Parameters for Updating a Device Class with a Device Type of DLT

##### *devclassname*

Specifies the name of the device class to be updated. This parameter is required.

##### **FORMAT=tapeformat**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

Possible values for DLT devices are:

##### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices are used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

##### **DLT10**

Specifies that ADSM writes data using the DLT10 recording format. This results in a tape capacity of approximately 10.0GB when using CompacTape III or CompacTape IV cartridges.

##### **DLT10C**

Specifies that ADSM writes data using the DLT10C recording format. This results in a tape capacity of approximately 10.0GB when using CompacTape III and CompacTape IV cartridges. This format, however, uses the tape drive's hardware compression feature, so the actual capacity may be greater than 10.0GB, depending on the effectiveness of the compression feature.

## UPDATE DEVCLASS

### **DLT20**

Specifies that ADSM writes data using the DLT20 recording format. This results in a tape capacity of approximately 20.0GB when using CompacTape IV cartridges.

**Attention:** The DLT20 format can only be used with CompacTape IV cartridges in a DLT4000 drive.

### **DLT20C**

Specifies that ADSM writes data using the DLT20C recording format. This results in a tape capacity of approximately 20.0GB when using CompacTape IV cartridges. This format, however, uses the tape drive's hardware compression feature, so the actual capacity of the cartridge may be greater than 20.0GB, depending on the effectiveness of the compression feature.

**Attention:** The DLT20C format can only be used with CompacTape IV cartridges in a DLT4000 drive.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the DLT drives that can be used by this device class. For information about defining a library object see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class.

If this value is set to 1, then it may not be possible for media-to-media operations (like reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name which the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

If you have already established a media label naming convention that supports your current management system, consider using a volume prefix that conforms to your naming conventions.

## UPDATE DEVCLASS

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You must specify this value as an integer followed by a scale factor. Possible scale factors are:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for DLT, see Table 163.

*Table 163. ADSM Default Estimated Capacity for DLT*

<b>Tape Format</b>	<b>Estimated Capacity</b>
DLT10	10.0GB
DLT10C	10.0GB (see note)
DLT20	20.0GB
DLT20C	20.0GB (see note)

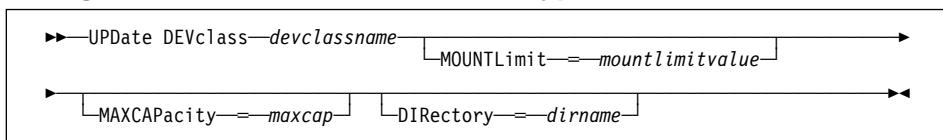
**Note:** This format uses the tape drive hardware compression feature, depending on the effectiveness of compression the actual capacity may be greater than the listed value.

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of FILE



### Parameters

#### Parameters for Updating a Device Class with a Device Type of FILE

##### *devclassname*

Specifies the name of the device class to be updated. This parameter is required.

##### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of files that can be simultaneously open for input/output. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive.

##### **MAXCAPacity**=*maxcap*

Specifies the maximum size of any data storage files categorized by this device class. This parameter is optional. You must specify this value as an integer followed by a scale factor. Possible scale factors are:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, MAXCAPACITY=5M specifies that the maximum capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, MAXCAPACITY=100K).

##### **DIRectory**=*dirname*

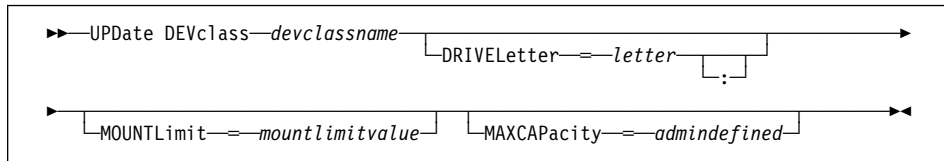
Specifies the directory location of the file or files used in this device class. This parameter is optional.

By specifying a directory name, you identify the location where the server places the files that represent storage volumes for this device class. While processing the command, the server expands the specified directory name into its fully qualified form, starting from the root directory.

Later, if the server needs to allocate a scratch volume, it creates a new file in this directory. For scratch volumes used to store client data, the file created by the server has a file name extension of .BFS. For scratch volumes used to store export data, a file name extension of .EXP is used. For example, suppose you define a device class with a DIRECTORY of **ADSMSTOR** and the server needs a scratch volume in this device class to store export data. The file which the server creates might then be named **ADSMSTOR\00566497.EXP**.

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of LAN3995



#### Parameters for Updating a Device Class with a Device Type of LAN3995

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### **DRIVELetter**=*letter*

Specifies the letter that identifies the drive. This parameter is optional.

##### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of volumes that can be simultaneously open for input/output. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class in the 3995 library.

##### **MAXCAPacity**=*maxcap*

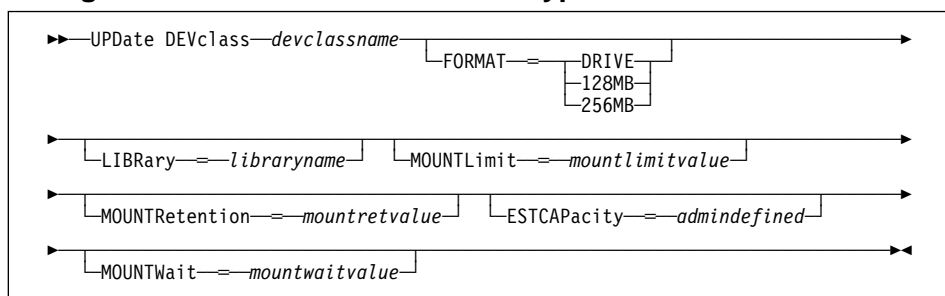
Specifies the maximum amount of data to store on each side of a volume in a storage pool associated with this device class. Since each volume has two sides, the maximum capacity of a volume is twice this value. This parameter is optional.

This value is specified as an integer followed by a scale factor. You must specify a scale factor following the specified integer. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of MO35



#### Parameters for Updating a Device Class with a Device Type of MO35

##### *devclassname*

Specifies the name of the device class to be defined. This parameter is required. The maximum length of the device class name is 30 characters.

##### **FORMAT=optformat**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

There are two possible values for optical devices:

##### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Warning:** Use the DRIVE value only when the device class has compatible sequential access devices. If you specify DRIVE for a device class that has incompatible sequential access devices, then you must mount volumes on devices that are capable of reading or writing at the format established when the volume is first mounted. This can cause errors if the only sequential access device that can access the volume is already in use.

##### **128MB**

Specifies that ADSM writes data to a 128MB three and one-half inch optical drive.

##### **256MB**

Specifies that ADSM writes data to a 256MB (230MB) three and one-half inch optical drive.

##### **LIBRARY=libraryname**

Specifies the name of the defined library object that contains the optical drives which can be used by this device class. This parameter is optional. For information about defining a library object, see the DEFINE LIBRARY command.

##### **MOUNTLIMIT=mountlimitvalue**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You

## UPDATE DEVCLASS

can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class.

If this value is set to 1, then it may not be possible for media-to-media operations (like reclamation) to work.

### **MOUNTRetention**=*mouretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

## UPDATE DEVCLASS

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes defined to a storage pool categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

This value is specified as an integer followed by a scale factor. You must specify a scale factor following the specified integer. For example, ESTCAPACITY=5M specifies that the estimated capacity for this device class is 5 megabytes. The following are possible scale factors:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

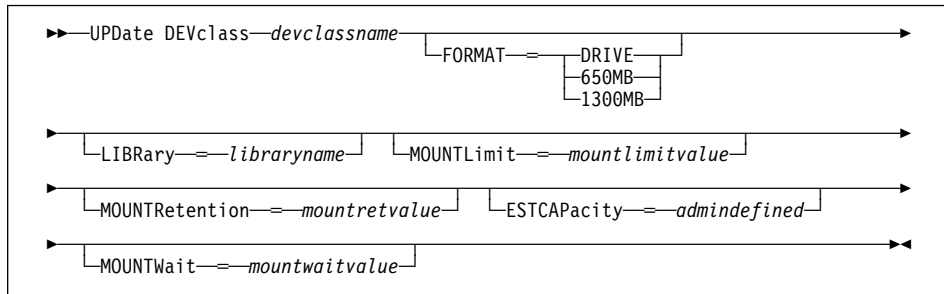
### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.



## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of OPTICAL



### Parameters for Updating a Device Class with a Device Type of OPTICAL

#### *devclassname*

Specifies the name of the device class to be defined. This parameter is required.

#### **FORMAT**=*optformat*

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

Possible values are:

#### **DRIVE**

Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the DRIVE value when a mixture of devices is used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option.

#### **650MB**

Specifies that ADSM writes data to a 650 MB 5.25-inch optical drive.

#### **1300MB**

Specifies that ADSM writes data to a 1300 MB 5.25-inch optical drive.

#### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the optical drives used by this device class. This parameter is optional. For information about defining a library object, see the DEFINE LIBRARY command.

#### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as, reclamation) to work.

## UPDATE DEVCLASS

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

<b>Symbol</b>	<b>Symbol Name</b>
K	Kilobyte
M	Megabyte
G	Gigabyte

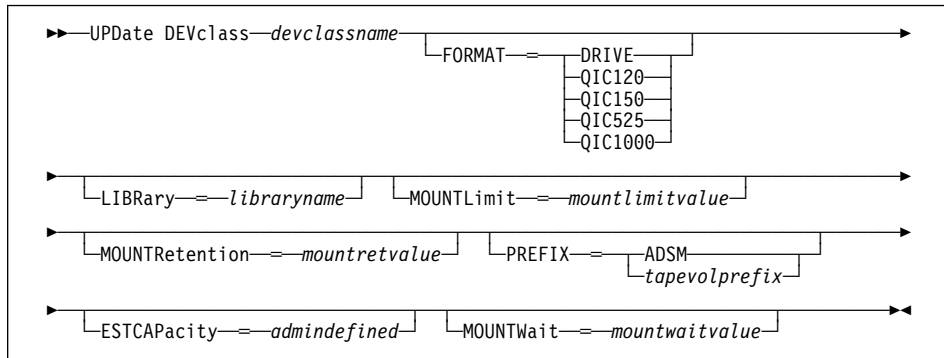
For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of QIC



### Parameters for Updating a Device Class with a Device Type of QIC

#### *devclassname*

Specifies the name of the device class to be defined. This parameter is required.

#### **FORMAT=tapeformat**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following are possible values for QIC devices:

**DRIVE** Specifies that ADSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted.

**Attention:** Avoid specifying the `DRIVE` value when a mixture of devices are used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the `FORMAT=DRIVE` option.

**QIC120** See Table 164 on page 568.

**QIC150** See Table 164 on page 568.

**QIC525** See Table 164 on page 568.

**QIC1000** See Table 164 on page 568.

## UPDATE DEVCLASS

Table 164. QIC Tape Recording Format Selections

Tape/Format	QIC-120	QIC-150	QIC-525	QIC-1000
3M DC300XLP	–	–	–	–
3M DC600A	Read	–	–	–
3M DC600XTD	Read/Write	Read/Write	–	–
3M DC6150	Read/Write	Read/Write	–	–
3M DC6320	Read/Write	Read/Write	Read/Write	–
3M DC6525	Read/Write	Read/Write	Read/Write	–
3M DC9100	–	–	–	Read/Write
3M DC9120XL	–	–	–	Read/Write

**Note:** ADSM cannot use 3M DC300XLP and 3M DC600A tapes.

### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the QIC tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### **MOUNTLimit**=*mountlimitvalue*

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive. This value should be equal to the number of drives available to this device class.

If this value is set to 1, then it may not be possible for media-to-media operations (like reclamation) to work.

### **MOUNTRetention**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### **PREFIX**=*tapevolprefix*

Specifies the prefix of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, ADSM uses this prefix to create the file name. This parameter is optional. The maximum length of this prefix is 8 characters.

If you have already established a media label naming convention that supports your current management system, consider using a volume prefix that conforms to your naming conventions.

### **ESTCAPacity**=*admindefined*

Specifies the estimated capacity for the sequential access volumes categorized by this device class. This parameter is optional. You can specify this parameter if the default estimated capacity for the device class is inaccurate due to compression of data.

## UPDATE DEVCLASS

You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 megabytes. The smallest value allowed is 100KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for QIC tapes, see Table 165.

*Table 165. ADSM Default Estimated Capacity for QIC Tape*

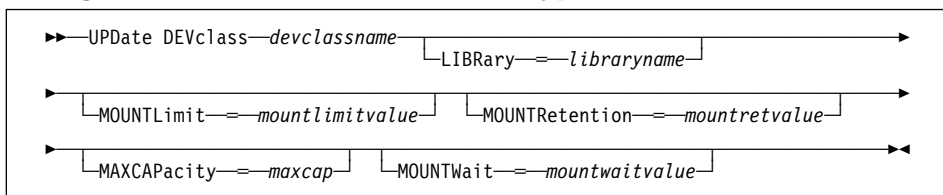
Tape Format	Estimated Capacity (range)
QIC120	26MB–172MB
QIC150	31MB–207MB
QIC525	65MB–427MB
QIC1000	169MB–1.1GB

### **MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.

## UPDATE DEVCLASS

### Syntax for Updating a Device Class with a Device Type of REMOVABLEFILE



#### Parameters for Updating a Device Class with a Device Type of REMOVABLEFILE

##### *devclassname*

Specifies the name of the device class to be updated. This parameter is required.

##### **LIBRARY**=*libraryname*

Specifies the name of the defined library object that contains the removable media drives used by this device class. This parameter is optional. For information about defining a library object, see the DEFINE LIBRARY command.

##### **MOUNTLIMIT**=*mountlimitvalue*

Specifies the maximum number of files that can be simultaneously open for input/output in this device class. This parameter is optional. You can specify an integer in the range of 1 to 256, inclusive.

Each removable medium contains only one file. It is recommended that this value is equal to the number of drives available to this device class.

If this value is set to 1, it may not be possible for media-to-media operations (such as reclamation) to work.

##### **MOUNTRETENTION**=*mountretvalue*

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify an integer in the range of 0 to 9999, inclusive.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

##### **MAXCAPACITY**=*maxcap*

Specifies the maximum size of any volumes (which are actually files) defined to a storage pool categorized by this device class. This parameter is optional. You must specify this value as an integer followed by a scale factor. The following are possible scale factors:

Symbol	Symbol Name
K	Kilobyte
M	Megabyte
G	Gigabyte

For example, MAXCAPACITY=5M specifies that the maximum capacity for a volume in this device class is 5 megabytes. The default value is 4MB (that is,

## UPDATE DEVCLASS

MAXCAPACITY=4M). The smallest value allowed is 100KB (that is, MAXCAPACITY=100K).

Because the server opens only one file per physical removable medium, specify a capacity such that the one file makes full use of your media capacity.

**MOUNTWait**=*mountwaitvalue*

Specifies the maximum length of time, in minutes, to wait for a sequential access volume to be mounted. This parameter is optional and does not apply to sequential access media libraries. If the mount request is not satisfied within the specified amount of time, the mount request is canceled. You can specify an integer in the range of 0 to 9999, inclusive.

### Examples

#### Task 1

Update a file device class named PLAINFILES to a maximum capacity of 25MB.

**Command:** `update devclass plainfiles maxcapacity=25m`

**Output:**

```
ANR2205I Device class PLAINFILES updated.
```

#### Task 2

Update an 8mm device class named 8MMTAPE to a 15 minute mount retention.

**Command:** `update devclass 8mmtape mountretention=15`

**Output:**

```
ANR2205I Device class 8MMTAPE updated.
```

## UPDATE DEVCLASS

### Related Commands

Table 166 provides references to related commands.

*Table 166. Commands Related to UPDATE DEVCLASS*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP DEVCONFIG	Backs up ADSM device information to a file.	Page 66
DEFINE DEVCLASS	Defines a device class that can be used for storage pools, importing, or exporting.	Page 120
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE DEVCLASS	Deletes a device class name.	Page 206
QUERY DEVCLASS	Displays information about one or more device classes.	Page 350



## UPDATE DOMAIN

### UPDATE DOMAIN (Update a Policy Domain)

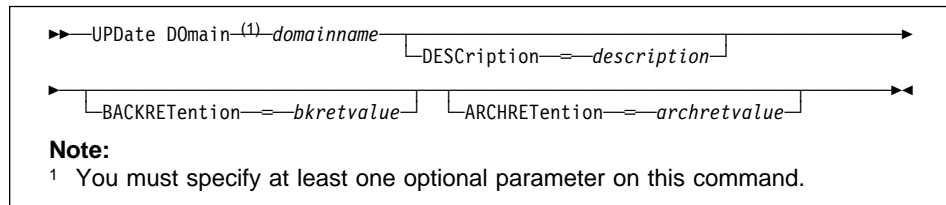
Use the UPDATE DOMAIN command to modify an existing policy domain.

You can use this command to update selected parameters for the specified policy domain. Any parameters that are not explicitly updated remain unchanged.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the specified policy domain.

### Syntax



### Parameters

#### *domainname*

Specifies the name of an existing policy domain to be updated. This parameter is required. The maximum length of this name is 30 characters.

#### **DESCRiption**=*description*

Describes the policy domain using a text string. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

#### **BACKRETention**=*bkretvalue*

Specifies the number of days (from the date of deactivation) to retain inactive backup versions of files when either of the following conditions are present:

- A file's management class is changed, but neither the new management class nor default management class contains a backup copy group.
- The management class to which a file is bound no longer exists, and the default management class does not contain a backup copy group.

Backup retention values can be in the range of 0 to 9999, inclusive.

#### **ARCHRETention**=*archretvalue*

Specifies the number of days (from the date of archive) to retain archive copies. The archive retention value is used by ADSM to manage copies of files when the management class to which a file is bound no longer exists, and the default management class does not contain an archive copy group. Backup retention values can be in the range of 0 to 9999, inclusive.

## UPDATE DOMAIN

### Examples

#### Task 1

Update the description of the EMPLOYEE\_RECORDS policy domain to Employee Information and set the backup retention parameter to 50 days.

**Command:** update domain employee\_records description="Employee Information" backretention=50

#### Output:

```
ANR1502I Policy domain EMPLOYEE_RECORDS updated.
```

#### Task 2

Update the description of the EMPLOYEE\_RECORDS policy domain to Development Programming Group. Set the backup retention parameter to 45 days and the archive retention parameter to 75 days.

**Command:** update domain employee\_records description="Development Programming Group" backretention=45 archretention=75

#### Output:

```
ANR1502I Policy domain EMPLOYEE_RECORDS updated.
```

### Related Commands

Table 167 provides references to related commands.

Table 167. Commands Related to UPDATE DOMAIN

Command	Description	Reference
COPY DOMAIN	Creates a copy of an existing policy domain.	Page 89
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE DOMAIN	Deletes a policy domain along with any policy objects in the policy domain.	Page 208
QUERY DOMAIN	Displays information about one or more policy domains.	Page 354

## UPDATE DRIVE

### UPDATE DRIVE (Update a Drive)

Use the UPDATE DRIVE command to update a drive that ADSM can use to access data on storage volumes. You typically use this command to redefine the device name during system reconfiguration.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
▶—UPDate DRive—libname—drivename—└─DEVIce—=devicename—┘
```

```
▶—└─ELEMEnt—=elementaddress—(1)—┘▶
```

**Note:**  
1 The ELEMENT parameter is invalid for drives in non-SCSI libraries.

### Parameters

#### *libname*

Specifies the name of the library to which the drive is assigned. This parameter is required.

#### *drivename*

Specifies the name that is assigned to the drive. This parameter is required.

#### **DEVIce**=*devicename*

Specifies the name, to be updated, of the device that corresponds to the drive. For example, you might specify TAPE05\$. This parameter is optional.

#### **ELEMEnt**=*elementaddress*

Specifies the new value to be used for the drive's element address within a SCSI library. This parameter is optional and is only valid for a drive located in a SCSI library. For a description of which element address values to use for various library configurations, see *ADSM Administrator's Guide*.

### Examples

#### Task

Update the drive named DRIVE3 located in the library named AUTO by changing the element address to 119.

## UPDATE DRIVE

**Command:** update drive auto drive3 element=119

**Output:**

```
ANR8467I Drive DRIVE3 in library AUTO updated.
```

### Related Commands

Table 168 provides references to related commands.

*Table 168. Commands Related to UPDATE DRIVE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DELETE DRIVE	Deletes a drive from a library.	Page 210
QUERY DRIVE	Displays information about one or more drives.	Page 359
QUERY LIBRARY	Displays information about one or more libraries.	Page 381

## UPDATE LIBRARY

---

### UPDATE LIBRARY (Update a Library)

Use the UPDATE LIBRARY command to modify a library device name or change the path name for an external library manager.

If you include the DEVCONFIG option in the dsmserv.opt file, the files you specify with that option are automatically updated with the results of this command. When you use this option, the files specified are automatically updated whenever a device class, library, or drive is defined, updated, or deleted.

### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax

```
►►—UPDate LIBRary—libraryname—DEVIce—=devicename—◄◄
```

### Parameters

*libraryname*

Specifies the name of the library to be updated. This parameter is required.

**DEVIce**=*devicename*

Specifies the new names of the devices that access the library's robotic mechanism. For example, you might specify AUTO02\$ as the device name. This parameter is required for SCSI, but is not valid for MANUAL library types. If the library type is SCSI, only one device name is allowed. For additional information, see the *ADSM Administrator's Guide*.

### Examples

#### Task

Update the library named AUTO with a new device name of AUTO02\$.

**Command:** update library auto device=auto02\$

#### Output:

```
ANR8465I Library AUTO updated.
```

## UPDATE LIBRARY

### Related Commands

Table 169 provides references to related commands.

*Table 169. Commands Related to UPDATE LIBRARY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
DEFINE DRIVE	Assigns a drive to a library.	Page 147
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
DELETE DRIVE	Deletes a drive from a library.	Page 210
DELETE LIBRARY	Deletes a library.	Page 220
QUERY DRIVE	Displays information about one or more drives.	Page 359
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
UPDATE DRIVE	Changes the attributes of a drive.	Page 575

## UPDATE LIBVOLUME

---

### UPDATE LIBVOLUME (Change the Status of a Storage Volume)

Use the UPDATE LIBVOLUME command to change the status of a sequential access storage volume in an existing library.

#### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

#### Syntax

```
►► Update LIBVolume—libraryname—volname—STATUS—  
└─ PRIVATE ─┘  
└─ SCRATCH ─┘
```

#### Parameters

##### *libraryname*

Specifies the name of the library where the storage volume is located. This parameter is required.

##### *volname*

Specifies the volume name of the storage volume. This parameter is required.

##### **STATUS=voltype**

Specifies a change to the state of a storage volume. This keyword is required. Possible values are:

##### **PRIVate**

Specifies that the storage volume is being updated to a private volume.

##### **SCRatch**

Specifies that the storage volume is being updated to a scratch volume.

#### Examples

##### Task

Update the volume named WPDV00 located in the library named AUTO to reflect a status of PRIVATE.

**Command:** update libvolume auto wpdv00 status=private

##### Output:

```
ANR8445I Volume WPDV00 in library AUTO updated.
```

## UPDATE LIBVOLUME

### Related Commands

Table 170 provides references to related commands.

*Table 170. Commands Related to UPDATE LIBVOLUME*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
AUDIT LIBRARY	Ensures that an automated library is in a consistent state.	Page 51
CHECKIN LIBVOLUME	Checks a storage volume into an automated library.	Page 80
CHECKOUT LIBVOLUME	Checks a storage volume out of an automated library.	Page 84
DEFINE LIBRARY	Defines an automated or manual library.	Page 150
QUERY LIBRARY	Displays information about one or more libraries.	Page 381
QUERY LIBVOLUME	Displays information about a library volume.	Page 384



## UPDATE MGMTCLASS

### UPDATE MGMTCLASS (Update a Management Class)

Use the UPDATE MGMTCLASS command to modify selected parameters for the specified management class. If you do not explicitly update a parameter, it remains unchanged.

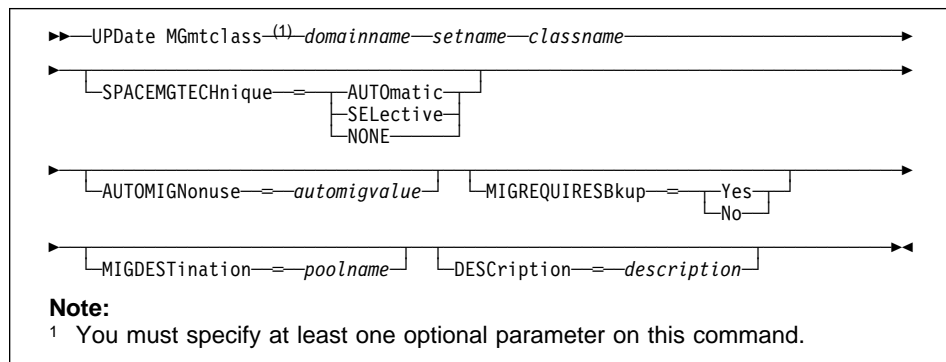
The following parameters affect only HSM clients that use this management class, and have no effect on backup-archive clients:

SPACEMGTECHNIQUE  
AUTOMIGNONUSE  
MIGREQUIRESBKUP  
MIGDESTINATION

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

### Syntax



### Parameters

#### *domainname*

Specifies the name of the policy domain to which the management class belongs. This parameter is required. The maximum length of this name is 30 characters.

#### *setname*

Specifies the name of the policy set to which the management class belongs. This parameter is required. The maximum length of this name is 30 characters. You cannot update a management class that belongs to the ACTIVE policy set.

#### *classname*

Specifies the name of the management class that you want to update. This parameter is required.

## UPDATE MGMTCLASS

### **SPACEMGTECHnique**=*smtvalue*

Specifies the space management technique to use for the management class. This parameter is optional. Possible values are:

#### **AUTOMATIC**

Specifies that the file is eligible for automatic migration or selective migration.

#### **SElective**

Specifies that the file is eligible for selective migration only.

#### **NONE**

Specifies that the file is not eligible for migration.

The default in the management class named STANDARD is NONE.

This parameter is effective only for HSM clients, not for backup-archive clients.

### **AUTOMIGNonuse**=*automigvalue*

Specifies the number of days that must elapse since the file was last used before it is eligible for automatic migration. This parameter is optional.

If the SPACEMGTECHNIQUE parameter is not AUTOMATIC, ADSM ignores this attribute. You can specify a value from 0–9999. The default value is 0.

This parameter is effective only for HSM clients, not for backup-archive clients.

### **MIGREQUIRESBkup**=*migreqvalue*

Specifies whether a backup version of a file must exist before a file can be migrated. This parameter is optional. Possible values are:

#### **Yes**

A backup version must exist.

#### **No**

A backup version is optional.

This parameter is effective only for HSM clients, not for backup-archive clients.

### **MIGDESTination**=*poolname*

Specifies the name of a primary storage pool that is the destination for migrated files. This parameter is optional. The default destination is SPACEMGPOOL.

A file must be associated with a management class that specifies an appropriate storage pool for migrated files. The type of storage devices in a storage pool and options specified for the storage pool affects the speed with which a migrated file can be recalled to a local file system.

The command fails if a copy storage pool is specified as the destination.

This parameter is effective only for HSM clients, not for backup-archive clients.

## UPDATE MGMTCLASS

### **DESCRiption**=*description*

Specifies a text string that describes the management class. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

### Examples

#### Task

Update the management class called ACTIVEFILES, for policy set VACATION in the EMPLOYEE\_RECORDS policy domain, with a description of Technical Design Support Mgmt Class.

**Command:** update mgmtclass employee\_records vacation activefiles  
description="Technical Design Support Mgmt Class"

#### Output:

```
ANR1522I Management class ACTIVEFILES updated in policy domain  
EMPLOYEE_RECORDS, set VACATION.
```

### Related Commands

Table 171 provides references to related commands.

Table 171 (Page 1 of 2). Commands Related to UPDATE MGMTCLASS

Command	Description	Reference
ASSIGN DEFMGMTCLASS	Assigns an existing management class as the default for a specified policy set.	Page 48
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157

## UPDATE MGMTCLASS

Table 171 (Page 2 of 2). Commands Related to UPDATE MGMTCLASS

Command	Description	Reference
DELETE MGMTCLASS	Deletes a management class and its copy groups from a policy domain and policy set.	Page 224
QUERY COPYGROUP	Displays the attributes of an existing copy group.	Page 334
QUERY MGMTCLASS	Displays information about one or more management classes.	Page 397
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539

## UPDATE NODE

### UPDATE NODE (Update a Client Node)

Use the UPDATE NODE command to modify the attributes of a registered client node.

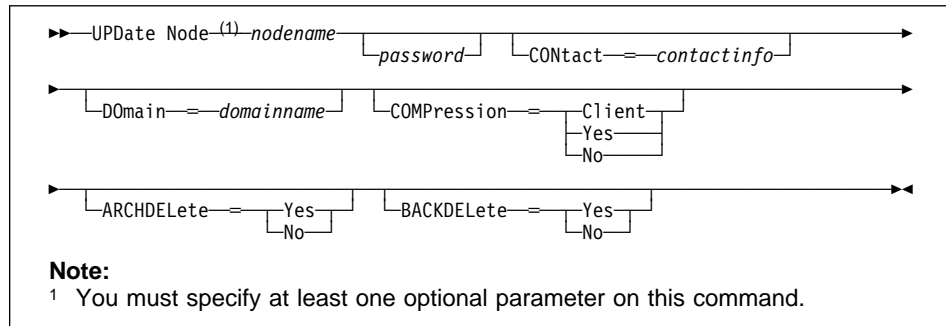
You can use this command to modify selected parameters for the specified client node. If you do not explicitly update a parameter, it remains unchanged.

You must use the RENAME NODE command to change the name of a registered client node.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the client node belongs.

### Syntax



### Parameters

#### *nodename*

Specifies the name of the client node to be updated. This parameter is required.

#### *password*

Specifies the new password for the client node. This parameter is optional. The maximum length of the name is 64 characters. Passwords remain current for a period determined by the password expiration period.

#### **CONTACT=***contactinfo*

Specifies a text string of information identifying the client node. This parameter is optional. The maximum length of the text string is 255 characters. The contact information must be enclosed in quotation marks if it contains any blanks. To remove previously defined contact information, specify a null string ("" ) for this value.

#### **DOMAIN=***domainname*

Specifies the name of the policy domain to which the client node is registered. This parameter is optional.

## UPDATE NODE

### **COMP**ression=*compressionvalue*

Specifies whether the client node compresses its files before sending them to the server for backup and archive. This parameter is optional. Possible values are:

#### **Client**

Specifies that the client determines whether files will be compressed.

#### **Yes**

Specifies that the client node compresses its files before sending them to the server for backup and archive.

#### **No**

Specifies that the client node does not compress its files before sending them to the server for backup and archive.

### **ARCHDE**lete=*archdeletevalue*

Specifies whether the client node can delete its own archived files from the server. This parameter is optional. Possible values are:

#### **Yes**

Specifies that the client node can delete its own archive files from the server.

#### **No**

Specifies that the client node cannot delete its own archive files from the server.

### **BACKDE**lete=*backdeletevalue*

Specifies whether the client node can delete its own backup files from the server. This parameter is optional. Possible values are:

#### **Yes**

Specifies that the client node can delete its own backup files from the server.

#### **No**

Specifies that the client node cannot delete its own backup files from the server.

## Examples

### **Task**

Update node LARRY so that the data on node LARRY will be compressed when it is backed up or archived by ADSM and so that the client cannot delete archived files.

**Command:** update node larry compression=yes archdelete=no

### **Output:**

```
ANR2063I Node LARRY updated.
```

## UPDATE NODE

### Related Commands

Table 172 provides references to related commands.

*Table 172. Commands Related to UPDATE NODE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY NODE	Displays partial or complete information about one or more clients.	Page 403
REGISTER NODE	Defines a client to the server and sets options for that user.	Page 466
REMOVE NODE	Removes a client from the list of registered nodes for a specific policy domain.	Page 472
RENAME NODE	Changes the name for an existing user that belongs to a specific policy domain.	Page 479

## UPDATE POLICYSET

---

### UPDATE POLICYSET (Update a Policy Set)

Use the UPDATE POLICYSET command to modify the description of an existing policy set.

You can use this command to modify the description for the specified policy set. If you do not explicitly update the DESCRIPTION parameter, it remains unchanged.

You cannot modify the description of the ACTIVE policy set.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

### Syntax

```
►►—UPDate POLIcYset—domainname—setname—DESCRiption=description—►►
```

### Parameters

*domainname*

Specifies the name of the policy domain to which the policy set belongs. This parameter is required.

*setname*

Specifies the name of the policy set to be updated. This parameter is required. You cannot modify the attributes of the ACTIVE policy set.

**DESCRiption**=*description*

Specifies a text string that describes the policy set. This parameter is required. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

### Examples

#### Task

Update a policy set called VACATION for the EMPLOYEE\_RECORDS policy domain with a description of Schedule Planning Information.

**Command:** update policyset employee\_records vacation description="Schedule Planning Information"

#### Output:

```
ANR1512I Policy set VACATION updated in policy domain  
EMPLOYEE_RECORDS.
```



## UPDATE POLICYSET

### Related Commands

Table 173 provides references to related commands.

*Table 173. Commands Related to UPDATE POLICYSET*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY MGMTCLASS	Creates a copy of an existing management class.	Page 92
DEFINE DOMAIN	Defines a policy domain that can be assigned to one or more clients.	Page 144
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DEFINE POLICYSET	Defines a policy set within the specified policy domain.	Page 161
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
VALIDATE POLICYSET	Verifies and reports on conditions the administrator must consider before activating the policy set.	Page 625

## UPDATE SCHEDULE

---

### UPDATE SCHEDULE (Update a Client or an Administrative Command Schedule)

Use the UPDATE SCHEDULE command to modify an existing schedule. The UPDATE SCHEDULE command takes two forms, depending on whether the schedule applies to client operations or administrative commands. The syntax and parameters for each form are defined separately.

#### Restrictions:

1. Not all clients can run all scheduled operations, even though ADSM allows you to define the schedule on the server and associate it with the client. For example, a Windows 3.1 client cannot run a schedule when the action is to restore or retrieve files, issue a command, or run an executable script (called a macro by ADSM). An executable script is also known as a command file, a batch file, or a script on different client operating systems. A Macintosh client cannot run a schedule when the action is to restore or retrieve files, or run an executable script.
2. You cannot schedule MACRO or QUERY ACTLOG commands.

---

### Updating Client Schedules

Use the UPDATE SCHEDULE command to modify selected parameters for the specified schedule. If you do not explicitly update a parameter, it remains unchanged.

This command does not change the client associations that have been made to this schedule. Any clients that are associated with the original schedule execute the modified schedule.

A schedule determines:

- The automatic function to be performed, for example, an incremental backup
- A startup window that indicates a time range during which the operation can begin
- The time period between startup windows

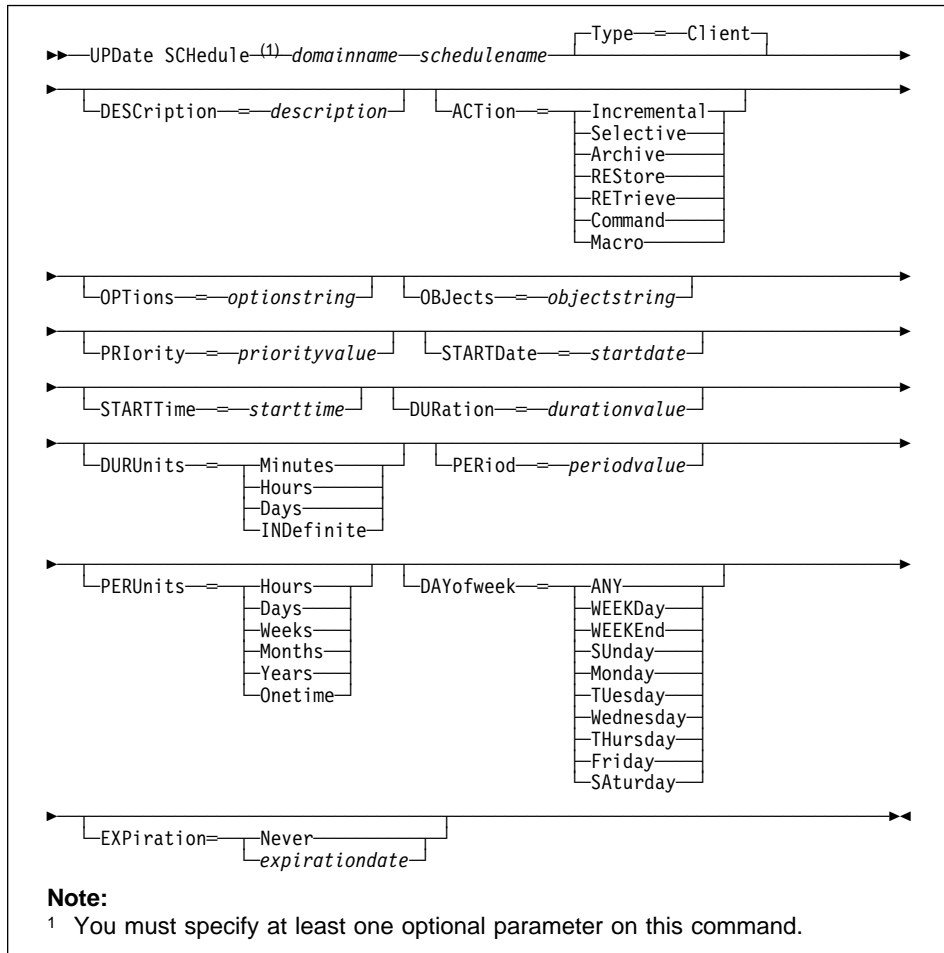
For each schedule, a startup window is specified. The startup window is the time period within which the scheduled operation must begin execution. The operation does not necessarily complete execution within this window. If the operation needs to be retried for any reason, the retry attempt must begin before the startup window elapses, or the operation is not retried.

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the schedule belongs.

## UPDATE SCHEDULE

### Syntax for Updating Client Schedules



### Parameters for Updating Client Schedules

#### *domainname*

Specifies the name of the policy domain to which this schedule belongs. This parameter is required.

#### *schedulename*

Specifies the name of the schedule to be updated. This parameter is required.

#### **Type=Client**

Specifies that this schedule updates a client schedule. This parameter is optional. The default value is CLIENT. Administrative command schedules are updated by specifying TYPE=ADMINISTRATIVE.

## UPDATE SCHEDULE

### **DESCRIPT***ion=description*

Specifies a description of the schedule. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

### **ACT***ion=actionvalue*

Specifies the action that occurs when this schedule is executed. This parameter is optional. Possible values are:

#### **Incremental**

Specifies that you want this schedule to initiate the backup of any files that have changed since the last backup and for which the specified time between backups has elapsed.

#### **Selective**

Specifies that you want this schedule to back up only the files specified with the OBJECTS parameter.

#### **Archive**

Specifies that you want to archive the files specified with the OBJECTS parameter.

#### **REStore**

Specifies that you want this schedule to restore the files specified with the OBJECTS parameter.

#### **RETri***e*

Specifies that you want this schedule to retrieve the files specified with the OBJECTS parameter.

#### **Command**

Specifies that the action is to be a client operating system command string specified with the OBJECTS parameter.

#### **Macro**

Specifies that a macro whose file name is specified with the OBJECTS parameter is to be executed.

### **OPT***ions=optionstring*

Specifies options that are supplied to the DSMC command at the time the schedule is executed. Only those options that are valid on the DSMC command line may be specified for this parameter. Refer to the appropriate client manual for information on which options are valid from the command line. This parameter is optional.

You can specify most options from the client's option file. The option string must be enclosed in quotation marks if it contains any blank characters. When applicable, these options override the options specified by a client after it has successfully contacted the server.

Do not include the following options because they have no effect on the client's execution of the scheduled command:

## UPDATE SCHEDULE

MAXCMDRETRIES  
QUERYSCHEDPERIOD  
RETRYPERIOD  
SCHEDLOGNAME  
SCHEDMODE  
TCPCLIENTADDRESS  
TCPCLIENTPORT

### **OBjects**=*objectstring*

Specifies the objects for which the specified action is performed. This parameter is optional. The allowed objects depend on the action you specify with the ACTION parameter and the operating system used by the clients associated with the schedule.

You can specify more than one object by separating the objects with commas, with no intervening spaces. For example, you can enter a string of file names or file spaces. If the value for this parameter contains any blank characters, you must enclose it in quotation marks.

When you specify ACTION=INCREMENTAL and do not specify any objects with this parameter, the DSMC command is invoked without specified objects and attempts to process objects as defined in the client's option file.

### **PRiority**=*priorityvalue*

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer from 1–10, inclusive, with 1 being the highest priority and 10 being the lowest.

If two or more schedules for a given client have the same window start time, this value determines the order of execution. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

### **STARTDate**=*startdate*

Specifies the date for the beginning of the window in which the schedule is first executed. This parameter is optional. This parameter is used in conjunction with the STARTTIME parameter to specify when the initial startup window begins.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the beginning of the startup window in MM/DD/YYYY format (for example, 01/31/1995).

### **TODAY**

Specifies that the beginning of the startup window is on the current date.

### **TODAY**+*numdays* or +*numdays*

Specifies that the beginning of the startup window is a specific number of days after the current date. The maximum value for the number of days is 9999.

## UPDATE SCHEDULE

For example, to specify that the beginning of the initial startup window is 3 days from now, you can specify `STARTDATE=TODAY+3` or `STARTDATE=+3`.

### **STARTTime**=*starttime*

Specifies the time for the beginning of the window in which the schedule is first executed, in HH:MM format. This parameter is optional. This parameter is used in conjunction with the `STARTDATE` parameter to specify when the startup window begins.

Possible values are:

#### *HH:MM:SS*

Specifies the time for the beginning of the startup window in HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that the beginning of the startup window is at the current time on the specified start date.

### **NOW**+*numhours:numminutes* **or** +*numhours:numminutes*

Specifies the time for the beginning of the startup window is the current time plus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the `UPDATE SCHEDULE` command at 5:00 with `STARTTIME=NOW+02:00` or `STARTTIME=+02:00`, the beginning of the initial startup window is at 7:00.

### **NOW**-*numhours:numminutes* **or** -*numhours:numminutes*

Specifies the time for the beginning of the startup window is at the current time minus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the `UPDATE SCHEDULE` command at 5:00 with `STARTTIME=NOW-02:00` or `STARTTIME=-02:00`, the beginning of the initial startup window is at 3:00.

### **DURation**=*durationvalue*

Specifies the number of units that define the length of the startup window. This parameter is optional. You can specify an integer from 1–999, inclusive.

This parameter is used in conjunction with the `DURUNITS` parameter to specify the length of the startup window. For example, if `DURATION=20` and `DURUNITS=MINUTES`, the schedule must be started within 20 minutes of the start date and start time. The duration of the window must be shorter than the period between windows.

This value is ignored if you specify `DURUNITS=INDEFINITE`.

### **DURUnits**=*durunitsvalue*

Specifies the time units used to determine the duration of the window in which this schedule can start. This parameter is optional.

## UPDATE SCHEDULE

This parameter is used in conjunction with the DURATION parameter to specify how long the startup window remains open to execute the schedule. For example, if DURATION=20 and DURUNITS=MINUTES, the schedule must be started within 20 minutes of the start date and start time. The duration of the window must be shorter than the period between windows. Possible values are:

### **Minutes**

Specifies that the duration of the window is defined in minutes.

### **Hours**

Specifies that the duration of the window is defined in hours.

### **Days**

Specifies that the duration of the window is defined in days.

### **INDefinite**

Specifies that the startup window has an indefinite duration. The schedule can run any time after the scheduled start time, until the schedule expires. You cannot specify DURUNITS=INDEFINITE unless PERUNITS=ONETIME is also specified for the schedule.

### **PERiod=*periodvalue***

Specifies the length of time between startup windows for this schedule. This parameter is optional. You can specify an integer from 1–999, inclusive.

This parameter is used in conjunction with the PERUNITS parameter to specify the period between startup windows. For example, if you specify PERIOD=5 and PERUNITS=DAYS (assuming that DAYOFWEEK=ANY), the operation is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window.

This value is ignored if you specify PERUNITS=ONETIME.

### **PERUnits=*perunitsvalue***

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional.

This parameter is used in conjunction with the PERIOD parameter to specify the period between startup windows. For example, if PERIOD=5 and PERUNITS=DAYS (assuming that DAYOFWEEK=ANY), the period between startup windows is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window. Possible values are:

### **Hours**

Specifies that time between startup windows for the schedule is specified in hours.

### **Days**

Specifies that time between startup windows for the schedule is specified in days.

## UPDATE SCHEDULE

### **Weeks**

Specifies that time between startup windows for the schedule is specified in weeks.

### **Months**

Specifies that time between startup windows for the schedule is specified in months.

### **Years**

Specifies that time between startup windows for the schedule is specified in years.

### **Onetime**

Specifies that the schedule executes only once. This value overrides the value specified by the PERIOD parameter.

### **DAYofweek=dayofweekvalue**

Specifies the day of the week on which the startup window begins. If the beginning of the calculated startup window falls on a day that does not correspond with your value for the DAYOFWEEK parameter, the startup window will be shifted forward in 24-hour increments until the DAYOFWEEK parameter is satisfied.

If you select a value for DAYOFWEEK other than ANY, and depending on the values for PERIOD and PERUNITS, schedules may not be executed when you would expect. Use the QUERY EVENT command to project when schedules will be executed and thereby ensure that the desired result is achieved.

Possible values are:

### **ANY**

Specifies that the startup window can begin on any day of the week.

### **WEEKDay**

Specifies that the startup window can begin on Monday, Tuesday, Wednesday, Thursday, or Friday.

### **WEEKEnd**

Specifies that the startup window can begin on Saturday or Sunday.

### **Sunday**

Specifies that the startup window begins on Sunday.

### **Monday**

Specifies that the startup window begins on Monday.

### **Tuesday**

Specifies that the startup window begins on Tuesday.

### **Wednesday**

Specifies that the startup window begins on Wednesday.

### **Thursday**

Specifies that the startup window begins on Thursday.



## UPDATE SCHEDULE

### **Friday**

Specifies that the startup window begins on Friday.

### **SAaturday**

Specifies that the startup window begins on Saturday.

### **EXpiration=expirationdate**

Specifies the date on which this schedule will no longer be used, in MM/DD/YYYY format. This parameter is optional. Possible values are:

### **Never**

Specifies that the schedule never expires.

### *expirationdate*

Specifies the date on which the schedule will no longer be used, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the specified date.

---

## Updating Administrative Command Schedules

Use the UPDATE SCHEDULE command to modify selected parameters for the specified schedule. If you do not explicitly update a parameter, it remains unchanged.

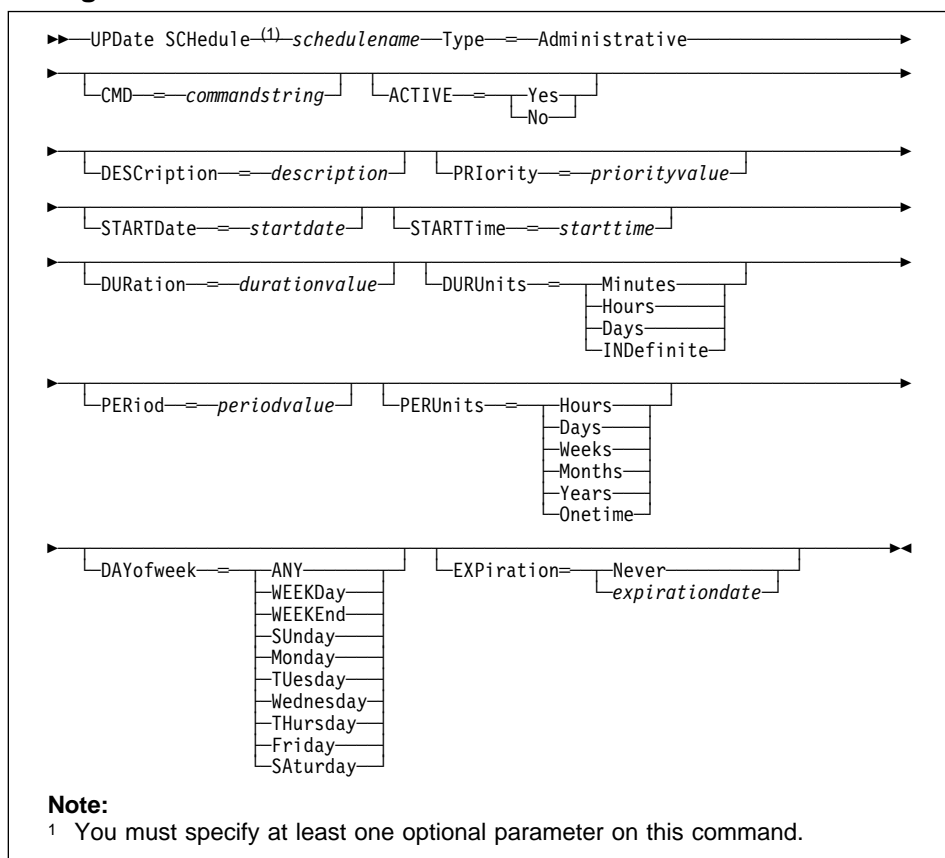
For each schedule, a startup window is specified. The startup window is the time period during which the schedule must be initiated. The schedule will not necessarily complete execution within this window. If the server is not running when this window starts but is started before the end of the defined window is reached, the schedule will run when the server is restarted.

## Privilege Class

To issue this command, you must have system privilege.

## UPDATE SCHEDULE

### Syntax for Updating Administrative Command Schedules



### Parameters for Updating Administrative Command Schedules

#### *schedulename*

Specifies the name of the schedule to be updated. This parameter is required.

#### **Type=Administrative**

Specifies that the schedule to be updated is an administrative schedule. This parameter is required for updating administrative command schedules.

#### **CMD=commandstring**

Specifies the actual administrative command that is to be scheduled. This parameter is optional. The maximum length of the command is 512 characters. The administrative command must be enclosed in quotation marks if it contains any blank characters.

#### **ACTIVE=activevalue**

Specifies whether the defined administrative command is eligible for scheduled execution. This parameter is optional. A defined administrative command

## UPDATE SCHEDULE

schedule will not be executed unless it is set into the active state. Possible values are:

**YES** Specifies that the defined administrative command is eligible for scheduled execution.

**NO** Specifies that the defined administrative command is not eligible for scheduled execution.

### **DESCRIPTION**=*description*

Specifies a description of the schedule. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

### **PRIORITY**=*priorityvalue*

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer from 1–10, inclusive, with 1 being the highest priority and 10 being the lowest.

If two or more schedules for a given client have the same window start time, this value determines the order of execution. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

### **STARTDATE**=*startdate*

Specifies the date for the beginning of the window in which the schedule is first executed. This parameter is optional. This parameter is used in conjunction with the STARTTIME parameter to specify when the initial startup window begins.

Possible values are:

*MM/DD/YYYY*

Specifies the date for the beginning of the startup window in MM/DD/YYYY format (for example, 01/31/1995).

### **TODAY**

Specifies that the beginning of the startup window is on the current date.

### **TODAY**+*numdays* or +*numdays*

Specifies that the beginning of the startup window is a specific number of days after the current date. The maximum value for the number of days is 9999.

For example, to specify that the beginning of the initial startup window is 3 days from now, you can specify STARTDATE=TODAY+3 or STARTDATE=+3.

### **STARTTIME**=*starttime*

Specifies the time for the beginning of the window in which the schedule is first executed, in HH:MM format. This parameter is optional. This parameter is used in conjunction with the STARTDATE parameter to specify when the startup window begins.

## UPDATE SCHEDULE

Possible values are:

### *HH:MM:SS*

Specifies the time for the beginning of the startup window in HH:MM:SS format (for example, 12:33:28).

### **NOW**

Specifies that the beginning of the startup window is at the current time on the specified start date.

### **NOW**+*numhours:numminutes* **or** +*numhours:numminutes*

Specifies the time for the beginning of the startup window is the current time plus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the UPDATE SCHEDULE command at 5:00 with STARTTIME=NOW+02:00 or STARTTIME=+02:00, the beginning of the initial startup window is at 7:00.

### **NOW**-*numhours:numminutes* **or** -*numhours:numminutes*

Specifies the time for the beginning of the startup window is at the current time minus a specific number of hours (and, optionally, minutes) on the specified start date.

For example, if you issue the UPDATE SCHEDULE command at 5:00 with STARTTIME=NOW-02:00 or STARTTIME=-02:00, the beginning of the initial startup window is at 3:00.

### **DURATION**=*durationvalue*

Specifies the number of units that define the length of the startup window. This parameter is optional. You can specify an integer in the range of 1–999, inclusive.

This parameter is used in conjunction with the DURUNITS parameter to specify the length of the startup window. For example, if DURATION=20 and DURUNITS=MINUTES, the schedule must be started within 20 minutes of the start date and start time. The duration of the window must be shorter than the period between windows.

This value is ignored if you specify DURUNITS=INDEFINITE.

### **DURUnits**=*durunitsvalue*

Specifies the time units used to determine the duration of the window in which this schedule can start. This parameter is optional.

This parameter is used in conjunction with the DURATION parameter to specify how long the startup window remains open to execute the schedule. For example, if DURATION=20 and DURUNITS=MINUTES, the schedule must be started within 20 minutes of the start date and start time. The duration of the window must be shorter than the period between windows. Possible values are:

### **Minutes**

Specifies that the duration of the window is defined in minutes.

## UPDATE SCHEDULE

### Hours

Specifies that the duration of the window is defined in hours.

### Days

Specifies that the duration of the window is defined in days.

### INDefinite

Specifies that the startup window has an indefinite duration. The schedule can run any time after the scheduled start time, until the schedule expires. You cannot specify DURUNITS=INDEFINITE unless PERUNITS=ONETIME is also specified for the schedule.

### PERiod=*periodvalue*

Specifies the length of time between startup windows for this schedule. This parameter is optional. You can specify an integer in the range of 1–999, inclusive.

This parameter is used in conjunction with the PERUNITS parameter to specify the period between startup windows. For example, if you specify PERIOD=5 and PERUNITS=DAYS (assuming that DAYOFWEEK=ANY), the operation is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window.

This value is ignored if you specify PERUNITS=ONETIME.

### PERUnits=*perunitsvalue*

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional.

This parameter is used in conjunction with the PERIOD parameter to specify the period between startup windows. For example, if PERIOD=5 and PERUNITS=DAYS (assuming that DAYOFWEEK=ANY), the period between startup windows is scheduled every 5 days after the initial start date and start time. The period between startup windows must exceed the duration of each window. Possible values are:

### Hours

Specifies that time between startup windows for the schedule is specified in hours.

### Days

Specifies that time between startup windows for the schedule is specified in days.

### Weeks

Specifies that time between startup windows for the schedule is specified in weeks.

### Months

Specifies that time between startup windows for the schedule is specified in months.

## UPDATE SCHEDULE

### **Years**

Specifies that time between startup windows for the schedule is specified in years.

### **Onetime**

Specifies that the schedule executes only once. This value overrides the value specified by the PERIOD parameter.

### **DAYofweek**=*dayofweekvalue*

Specifies the day of the week on which the startup window begins. If the beginning of the calculated startup window falls on a day that does not correspond with your value for the DAYOFWEEK parameter, the startup window will be shifted forward in 24-hour increments until the DAYOFWEEK parameter is satisfied.

If you select a value for DAYOFWEEK other than ANY, depending on the values for PERIOD and PERUNITS, schedules may not be executed when you would expect. Use the QUERY EVENT command to project when schedules will be executed and thereby ensure that the desired result is achieved. Possible values are:

### **ANY**

Specifies that the startup window can begin on any day of the week.

### **WEEKDay**

Specifies that the startup window can begin on Monday, Tuesday, Wednesday, Thursday, or Friday.

### **WEEKEnd**

Specifies that the startup window can begin on Saturday or Sunday.

### **SUNday**

Specifies that the startup window begins on Sunday.

### **Monday**

Specifies that the startup window begins on Monday.

### **Tuesday**

Specifies that the startup window begins on Tuesday.

### **Wednesday**

Specifies that the startup window begins on Wednesday.

### **Thursday**

Specifies that the startup window begins on Thursday.

### **Friday**

Specifies that the startup window begins on Friday.

### **SAaturday**

Specifies that the startup window begins on Saturday.

## UPDATE SCHEDULE

### **EXPIRATION=expirationdate**

Specifies the date on which this schedule will no longer be used, in MM/DD/YYYY format. This parameter is optional. Possible values are:

#### **Never**

Specifies that the schedule never expires.

#### *expirationdate*

Specifies the date on which the schedule will no longer be used, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the specified date.

## Examples

### **Task 1**

Update the MONTHLY\_BACKUP schedule that belongs to the STANDARD policy domain by setting its priority value to 1.

**Command:** update schedule standard monthly\_backup priority=1

#### **Output:**

```
ANR2502I Schedule MONTHLY_BACKUP updated in policy domain
STANDARD.
```

### **Task 2**

Update the WEEKLY\_BACKUP schedule that belongs to the EMPLOYEE\_RECORDS policy domain to expire on March 29, 1996 (03/29/1996).

**Command:** update schedule employee\_records weekly\_backup  
expiration=03/29/1996

#### **Output:**

```
ANR2502I Schedule WEEKLY_BACKUP updated in policy domain EMPLOYEE_RECORDS.
```

### **Task 3**

Update existing administrative schedule named BACKUP\_BACKUPPOOL so that starting today, the BACKUPPOOL primary storage pool is backed up to the COPYSTG copy storage pool every three days at 10:00 p.m.

**Command:** update schedule backup\_backuppool type=administrative cmd="backup  
stgpool backuppool copystg" active=yes starttime=22:00 period=3

## UPDATE SCHEDULE

### Output:

```
ANR2532I Schedule BACKUP_BACKUPPOOL updated.
```

### Related Commands

Table 174 provides references to related commands.

*Table 174. Commands Related to UPDATE SCHEDULE*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
COPY SCHEDULE	Creates a copy of an existing schedule.	Page 96
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.	Page 164
DELETE SCHEDULE	Deletes a schedule from the database.	Page 229
QUERY EVENT	Displays information about scheduled and completed events for selected clients.	Page 362
QUERY SCHEDULE	Displays information about one or more schedules.	Page 422
SET MAXCMDRETRIES	Specifies the maximum number of retries after a failed attempt to execute a scheduled command.	Page 517
SET RETRYPERIOD	Specifies the time between retry attempts by the client scheduler.	Page 530



## UPDATE STGPOOL

### UPDATE STGPOOL (Change the Attributes of a Storage Pool)

Use the UPDATE STGPOOL command to change any parameters in an existing storage pool.

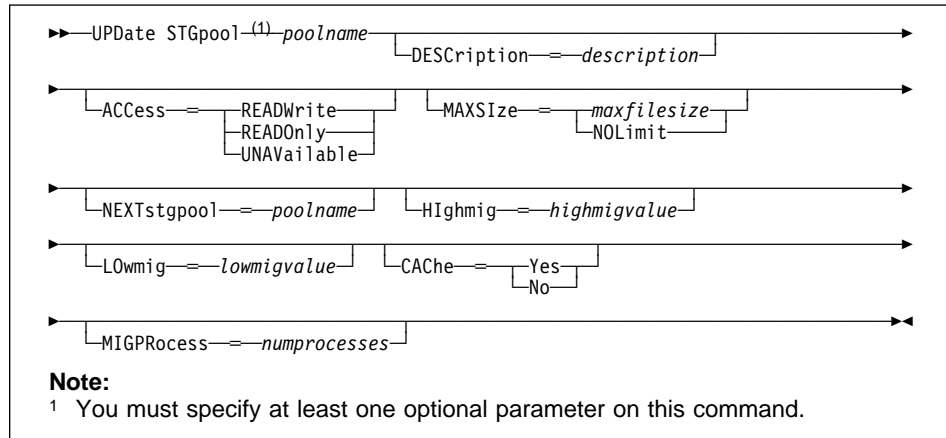
You can use this command to modify selected parameters for the specified storage pool. If you do not explicitly update a parameter, it remains unchanged.

The UPDATE STGPOOL command takes three forms, depending on whether the update is for a primary storage pool assigned to random access devices, a primary storage pool assigned to sequential access devices, or a copy storage pool. The syntax and parameters for each form are defined separately.

### Privilege Class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool to be updated.

### Syntax for Updating a Random Access Storage Pool



### Parameters for Updating a Random Access Storage Pool

*poolname*

Specifies the name of the storage pool to be updated. This parameter is required.

**DESCRIPTION**=*description*

Specifies a description of the storage pool to be updated. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

## UPDATE STGPOOL

### **ACCess**=*accessmode*

Specifies how users and system processes (like migration and reclamation) can access files in the storage pool. This parameter is optional. Possible values are:

#### **READWrite**

Specifies that users and system processes (like migration) can read and write to files stored on volumes in the storage pool.

#### **READOnly**

Specifies that users can only read files stored on the volumes in the storage pool.

No new writes are permitted to the volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *readonly*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool from volumes in another storage pool.

#### **UNAVailable**

Specifies that users cannot access files stored on volumes in the storage pool.

No new writes are permitted to the volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool. Files can also be moved or copied from this storage pool to another storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *unavailable*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool.

### **MAXSize**=*maxfilesize*

Specifies the maximum file size for the storage pool. This parameter is optional. The value of this parameter can be either NOLIMIT, or an integer followed by a scale factor. The value NOLIMIT specifies that there is no maximum size limit for files stored in the storage pool.

To limit the maximum file size, specify an integer (1–999999) followed by a scale factor. Scale factors and their meanings are:

<b>Scale Factor</b>	<b>Meaning</b>
K	Kilobyte
M	Megabyte
G	Gigabyte
T	Terabyte

## UPDATE STGPOOL

For example, `MAXSIZE=5G` specifies that the maximum file size for this storage pool is 5 gigabytes.

If a file exceeds the maximum size and no pool is specified as the `NEXTSTGPOOL`, the file is not stored. If a file exceeds the maximum size and a pool is specified as the `NEXTSTGPOOL`, the file is stored in the next storage pool that can accept the file size. If you specify the `NEXTSTGPOOL` parameter, at least one storage pool in the hierarchy should have a `MAXSIZE` value of `NOLIMIT`. By setting `MAXSIZE` to `NOLIMIT` for at least one pool, you ensure that no matter what its size, a file can be stored somewhere.

### **NEXTstgpool=***poolname*

Specifies the name of the storage pool to which files can be migrated. This parameter is optional. The pool specified must be a primary storage pool. A copy storage pool cannot be specified.

To remove an existing storage pool from the storage hierarchy, specify a null string ("") for this value.

If you do not specify a value for this parameter, migration cannot occur and files that exceed the value specified for the `MAXSIZE` parameter cannot be placed in another storage pool.

You cannot create a chain of storage pools that leads to an endless loop through the `NEXTSTGPOOL` parameter. At least one storage pool in the hierarchy must have no value specified for `NEXTSTGPOOL`.

### **Highmig=***highmigvalue*

Specifies when ADSM can start migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 0–100, inclusive.

When the storage pool occupancy exceeds the value for this parameter, ADSM can start migration, by node, to the next storage pool, as defined with the `NEXTSTGPOOL` parameter. You can specify `HIGHMIG=100` to prevent migration for this storage pool (assuming you have defined the `NEXTSTGPOOL` parameter).

### **LOwmig=***lowmigvalue*

Specifies when ADSM can stop migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 0–99, inclusive.

When the storage pool occupancy reaches the value for this parameter, ADSM does not initiate migration of another node's files. Because all file spaces that belong to a node are migrated together, the occupancy of the storage pool can fall below the value you specified for this parameter. You can set `LOWMIG=0` to permit migration to empty the storage pool.

### **CAChe=***cachingvalue*

For random access storage pools only, specifies whether a cached copy of a file that has been migrated remains in the random access storage pool. This

## UPDATE STGPOOL

parameter is optional. The primary copy of the file is migrated to the next storage pool, as defined with the NEXTSTGPOOL parameter. Caching improves the time it takes to retrieve files that have been migrated to sequential access storage pools by permitting cached copies of files to remain in the random access storage pool. The cached copies are available, if needed, or can be overwritten when necessary. Possible values are:

### Yes

Specifies that caching is enabled.

### No

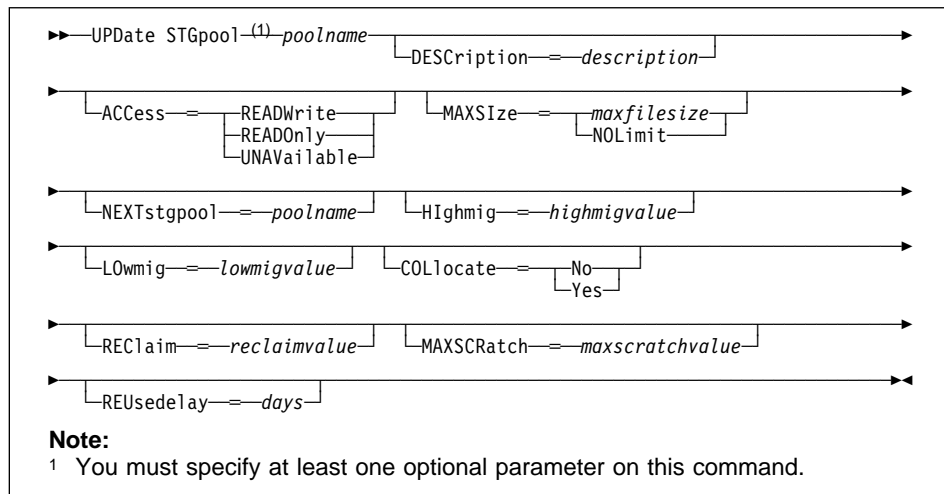
Specifies that caching is disabled.

### MIGPRocess=*numprocesses*

For random access storage pools only, specifies the number of processes that are used for migrating files from this storage pool. This parameter is optional. You can specify an integer from 1–999, inclusive.

During migration, these processes are performed in parallel to provide the potential for improved migration rates.

## Syntax for Updating a Primary Sequential Access Storage Pool



## Parameters for Updating a Primary Sequential Access Storage Pool

### *poolname*

Specifies the name of the storage pool to be updated. This parameter is required.

### DESCRIPTION=*description*

Specifies a description of the storage pool to be updated. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank

## UPDATE STGPOOL

characters. To remove a previously defined description, specify a null string ("") for this value.

### **ACCess**=*accessmode*

Specifies how users and system processes (like migration and reclamation) can access files in the storage pool. This parameter is optional. Possible values are:

#### **READWrite**

Specifies that users and system processes (like migration) can read and write files stored on volumes in the storage pool.

#### **READOnly**

Specifies that users can only read files stored on the volumes in the storage pool.

No new writes are permitted to the volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *readonly*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool from volumes in another storage pool.

#### **UNAVailable**

Specifies that users cannot access files stored on volumes in the storage pool.

No new writes are permitted to the volumes in the storage pool from other volumes outside the storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool. Files can also be moved or copied from this storage pool to another storage pool.

If this storage pool has been specified as a subordinate storage pool (with the NEXTSTGPOOL parameter) and is defined as *unavailable*, the storage pool is skipped when system processes (like migration) attempt to write files to the storage pool.

### **MAXSize**=*maxfilesize*

Specifies the maximum file size for the storage pool. This parameter is optional. The value of this parameter can be either NOLIMIT, or an integer followed by a scale factor. The value NOLIMIT specifies that there is no maximum size limit for files stored in the storage pool.

To limit the maximum file size, specify an integer (1–999999) followed by a scale factor. Scale factors and their meanings are:

<b>Scale Factor</b>	<b>Meaning</b>
K	Kilobyte
M	Megabyte
G	Gigabyte
T	Terabyte

## UPDATE STGPOOL

For example, `MAXSIZE=5G` specifies that the maximum file size for this storage pool is 5 gigabytes.

If a file exceeds the maximum size and no pool is specified as the `NEXTSTGPOOL`, the file is not stored. If a file exceeds the maximum size and a pool is specified as the `NEXTSTGPOOL`, the file is stored in the next storage pool that can accept the file size. If you specify the `NEXTSTGPOOL` parameter, at least one storage pool in the hierarchy should have a `MAXSIZE` value of `NOLIMIT`. By setting `MAXSIZE` to `NOLIMIT` for at least one pool, you ensure that no matter what its size, a file can be stored somewhere.

### **NEXTstgpool**=*poolname*

Specifies the name of the storage pool to which files can be migrated. This parameter is optional. The storage pool must be a primary storage pool. A copy storage pool cannot be specified.

To remove an existing storage pool from the storage hierarchy, specify a null string ("") for this value.

If you do not specify a value for this parameter, migration cannot occur and files that exceed the value specified for the `MAXSIZE` parameter cannot be placed in another storage pool.

You cannot create a chain of storage pools that leads to an endless loop through the `NEXTSTGPOOL` parameter. At least one storage pool in the hierarchy must have no value specified for `NEXTSTGPOOL`.

### **Highmig**=*highmigvalue*

Specifies when ADSM can start migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 1–100, inclusive.

When the storage pool occupancy exceeds the value for this parameter, ADSM can start migration, by volume, to the next storage pool, as defined with the `NEXTSTGPOOL` parameter. You can specify `HIGHMIG=100` to prevent migration for this storage pool (assuming you have defined the `NEXTSTGPOOL` parameter).

### **Lowmig**=*lowmigvalue*

Specifies when ADSM can stop migration for this storage pool, as a percentage of the storage pool occupancy. This parameter is optional. You can specify an integer from 0–99, inclusive.

When the storage pool occupancy reaches the value for this parameter, ADSM does not initiate migration of files from another volume. Because all files that are on a single volume are migrated together, the occupancy of the storage pool can fall below the value you specified for the `LOWMIG` parameter. You can set `LOWMIG=0` to permit migration to empty the storage pool.

### **COLlocate**=*collocation*

Specifies whether ADSM attempts to keep file spaces assigned to as few volumes as possible. This parameter is optional.

## UPDATE STGPOOL

Collocation reduces the number of sequential access media mounts for restore, retrieve, and recall operations. However, collocation increases both the amount of server time needed to collocate files for storing, and the number of volumes required. Collocation may not be useful on systems with only one or two sequential devices. For additional information, see *ADSM Administrator's Guide*.

Possible values are:

**Yes**

Specifies that collocation is enabled.

**No**

Specifies that collocation is disabled.

**REClaim**=*reclaimvalue*

Specifies when to initiate reclamation of fragmented space on volumes in the storage pool, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space usable again by moving any remaining active files from one volume to another volume, thus making the original volume available for reuse. This parameter is optional. You can specify an integer from 1–100, inclusive.

Space becomes fragmented as files on a volume are deleted because of processes such as expiration or file space deletion. The amount of space that can be reclaimed on a volume increases as files on the volume are deleted. When the percentage of space that can be reclaimed on a volume rises above the value for the RECLAIM parameter, the volume is reclaimed. Active files on the volume are rewritten to other volumes in the storage pool, making the original volume available for new files.

For storage pools, it is recommended that you specify a value of 50 percent or greater for this parameter so that files stored on two volumes can be combined onto a single output volume.

**MAXSCRatch**=*maxscratchvalue*

Specifies the maximum number of scratch volumes that can be requested by ADSM. This parameter is optional. You can specify an integer from 0–100000000, inclusive.

If you do not specify a value for this parameter, no scratch media are requested by ADSM.

Scratch volumes are automatically deleted from the storage pool when they become empty. When scratch volumes with the device type of FILE are deleted, the space that the volumes occupied is freed by ADSM and returned to the file system.

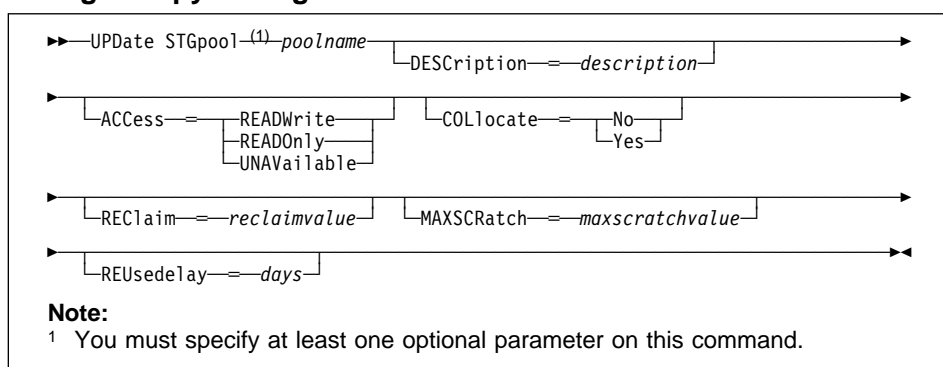
**REUsedelay**=*days*

Specifies the number of days that must elapse after all files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. This parameter is optional. You can specify an integer from 0–9999, inclusive. The value 0 means that a volume can be rewritten or returned to the scratch pool as soon as all files have been deleted from the volume.

## UPDATE STGPOOL

By specifying this parameter, you can ensure that the database could be restored to an earlier level and database references to files in the storage pool would still be valid.

### Syntax for Updating a Copy Storage Pool



### Parameters for Updating a Copy Storage Pool

#### *poolname*

Specifies the name of the copy storage pool to be updated. This parameter is required.

#### **DESCRiption**=*description*

Specifies a description of the copy storage pool to be updated. This parameter is optional. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("") for this value.

#### **ACCess**=*accessmode*

Specifies how users and system processes (like reclamation) can access files in the copy storage pool. This parameter is optional. Possible values are:

##### **READWrite**

Specifies that files can be read from and written to the volumes in the copy storage pool.

##### **READOnly**

Specifies that users can only read files stored on the volumes in the copy storage pool.

No new writes are permitted to the volumes in the copy storage pool from other volumes outside the copy storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the storage pool. Files cannot be backed up to the copy storage pool, but existing backup copies can be restored.



## UPDATE STGPOOL

### **UNAVailable**

Specifies that users cannot access files stored on volumes in the copy storage pool.

No new writes are permitted to the volumes in the copy storage pool from other volumes outside the copy storage pool. However, system processes (like reclamation) are permitted to move files within the volumes in the copy storage pool. Files cannot be backed up to the copy storage pool, but existing backup copies can be restored.

### **COLlocate=collocation**

Specifies whether ADSM attempts to keep file spaces assigned to as few volumes as possible. This parameter is optional.

Collocation reduces the number of sequential access media mounts for restore, retrieve, and recall operations. However, collocation increases both the amount of server time needed to collocate files for storing, and the number of volumes required. Collocation may not be useful on systems with only one or two sequential devices. For additional information, see *ADSM Administrator's Guide*.

Possible values are:

#### **Yes**

Specifies that collocation is enabled.

#### **No**

Specifies that collocation is disabled.

### **REClaim=reclaimvalue**

Specifies when to initiate reclamation of fragmented space on volumes in the copy storage pool, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space usable again by moving any remaining active files from one volume to another volume, thus making the original volume available for reuse. This parameter is optional. You can specify an integer from 1–100, inclusive. The value 100 means that reclamation is not performed.

The amount of space that can be reclaimed on a volume increases as files on the volume are deleted because of processes such as expiration or file space deletion. It is recommended that you specify a value of 50 percent or greater for this parameter so that files stored on two volumes can be combined onto a single output volume.

When reclamation occurs and how reclamation processing is done depends on whether the volumes are marked as offsite.

For volumes that are not marked as offsite volumes, reclamation can occur only after the volume has been filled, and then begins to empty because of file deletion. When the percentage of reclaimable space on a volume that had once been full rises above the value for the RECLAIM parameter, the volume is reclaimed. Active files on the volume are rewritten to other volumes in the storage pool, making the original volume available for new files.

For offsite volumes, reclamation can occur regardless of whether the volume has ever been filled. An offsite volume is eligible for reclamation when the

## UPDATE STGPOOL

percentage of unused space on the volume is greater than the reclaim parameter value. The unused space includes both space that has never been used on the volume and space that has become empty because of file deletion.

Reclamation processing also works differently for offsite storage pool volumes. When a copy pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to retrieve the active files on the reclaimable volume from a primary or copy storage pool that is onsite, and then write these files to an available volume in the original copy storage pool. Effectively, these files are moved back to the onsite location, although they could be obtained from the offsite volume after a disaster if a database backup is used which references the files on the offsite volume. Because of the way reclamation works with offsite volumes, use it carefully with copy storage pools.

### **MAXSCRatch**=*maxscratchvalue*

Specifies the maximum number of scratch volumes that can be requested by ADSM. This parameter is optional. You can specify an integer from 0–100000000, inclusive.

If you do not specify a value for this parameter, no scratch media are requested by ADSM.

A scratch volume is automatically deleted from the copy pool after all files have been deleted and the time specified by the REUSEDELAY parameter has elapsed. However, if the access mode for a scratch volume is OFFSITE, the volume is not deleted from the copy pool until the access mode is changed; this allows an administrator to query ADSM for empty, offsite scratch volumes and return these to the onsite location.

When scratch volumes with the device type of FILE become empty and are deleted, the space that the volumes occupied is freed by ADSM and returned to the file system.

### **REUsedelay**=*days*

Specifies the number of days that must elapse after all files have been deleted from a volume, before the volume can be rewritten or returned to the scratch pool. This parameter is optional. You can specify an integer from 0–9999, inclusive. A value of 0 means that a volume can be rewritten or returned to the scratch pool as soon as all files have been deleted from the volume.

**Important:** By specifying this parameter, you can ensure that the database can be restored to an earlier level and database references to files in the copy storage pool are still valid. For this to happen, you must set this parameter to a value greater than the number of days you plan on retaining the oldest database backup. For additional information, refer to *ADSM Administrator's Guide*.

## Examples

### Task 1

Update the primary storage pool named 8MMPOOL to permit as many as 10 scratch volumes to be mounted.

## UPDATE STGPOOL

**Command:** update stgpool 8mmpool maxscratch=10

**Output:**

```
ANR2202I Storage pool 8MMP00L updated.
```

### Task 2

Update the primary storage pool named BACKUPPOOL to permit caching.

**Command:** update stgpool backuppool cache=yes

**Output:**

```
ANR2202I Storage pool BACKUPPOOL updated.
```

### Task 3

Update the copy storage pool named COPYSTG to permit as many as 10 scratch volumes to be mounted and to collate files.

**Command:** update stgpool copystg collocate=yes maxscratch=10

**Output:**

```
ANR2202I Storage pool COPYSTG updated.
```

## Related Commands

Table 175 provides references to related commands.

*Table 175 (Page 1 of 2). Commands Related to UPDATE STGPOOL*

Command	Description	Reference
BACKUP DB	Backs up the ADSM database to sequential access volumes.	Page 61
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.	Page 68
DEFINE STGPOOL	Defines a storage pool as a named collection of data storage media.	Page 180
DELETE STGPOOL	Deletes a storage pool from data storage.	Page 232

## UPDATE STGPOOL

*Table 175 (Page 2 of 2). Commands Related to UPDATE STGPOOL*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY STGPOOL	Displays information about one or more storage pools.	Page 436
RESTORE STGPOOL	Restores files to a primary storage pool from copy storage pools.	Page 488
RESTORE VOLUME	Restores files stored on specified volumes in a primary storage pool from copy storage pools.	Page 493

## UPDATE VOLHISTORY

---

### UPDATE VOLHISTORY (Add the Location of a Database Dump, Database Backup, or Export Volume)

Use the UPDATE VOLHISTORY command to add volume location information to volume history files. You can use this location information to help track volumes that move offsite. The physical location information can be added only for a volume that contains the results of a database dump, a database backup (full or incremental), or an export operation. The location is added to the volume history files as a comment above the specified volume's detail information.

#### Privilege Class

To issue this command, you must have system privilege or unrestricted storage privilege.

#### Syntax

```
►—UPDate VOLHistory—volumename—DEVclass—=devclassname—►
►—LOcation—=location—►
```

#### Parameters

##### *volumename*

Specifies the name of the volume for which you want to add location information. The volume must have been used for a database dump, a database backup (full or incremental), or an export operation. This parameter is required.

##### DEVclass=*devclassname*

Specifies the name of the device class for the volume specified. This parameter is required.

##### LOcation=*location*

Describes the location of the volume. This parameter is required. The maximum length of the location is 255 characters. The location must be enclosed in quotation marks if it contains any blank characters.

#### Examples

##### Task

Update the location of a volume used for database backup, BACKUP1, to show that it has been moved to an offsite location.

**Command:** update volhistory backup1 devclass=tapebkup location="700 W. Magee Rd."

##### Output:

```
ANR2473I UPDATE VOLHISTORY for volume backup1 completed.
```

## UPDATE VOLHISTORY

### Related Commands

Table 176 provides references to related commands.

*Table 176. Commands Related to UPDATE VOLHISTORY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
BACKUP VOLHISTORY	Records sequential volume history information in external files.	Page 72
DELETE VOLHISTORY	Removes sequential volume history information from the volume history file.	Page 234
QUERY VOLHISTORY	Displays sequential volume history information that has been collected by the server.	Page 443

## UPDATE VOLUME

---

### UPDATE VOLUME (Change the Attributes of One or More Storage Pool Volumes)

Use the UPDATE VOLUME command to change the access mode for one or more volumes in random or sequential access storage pools. You can also use this command to change the location information for one or more volumes in sequential access storage pools. This command can be used to update volumes assigned to primary or copy storage pools. If you do not explicitly update a parameter, it remains unchanged.

You can reset any error state associated with a volume by updating the volume to an access mode of READWRITE.

A random access volume must be varied offline before you can update the volume to *unavailable* or *destroyed*. To vary a volume offline, use the VARY command.

If a scratch volume with a status of EMPTY and an access mode of OFFSITE is updated so that the access mode is READWRITE, READONLY, or UNAVAILABLE, the volume is deleted from the database.

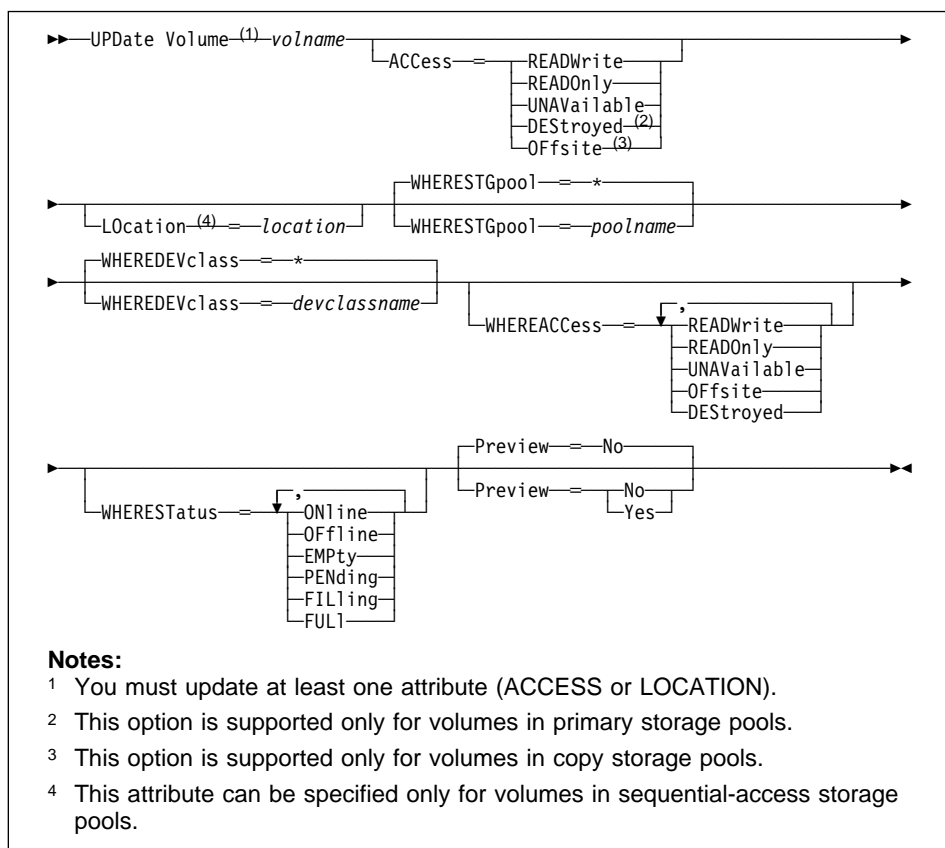
Use the QUERY VOLUME command to display the set of defined storage pool volumes.

### Privilege Class

To issue this command, you must have system privilege or operator privilege.

## UPDATE VOLUME

### Syntax



### Parameters

#### *volname*

Specifies name of the storage pool volume to be updated. This parameter is required; however, you can use a pattern-matching expression to specify this name. All matching volumes are included in the update.

#### **ACCESS**=*newaccessmode*

Specifies how users and system processes (like migration) can access files in the storage pool volume. The access mode of all updated volumes will be changed to this value. This parameter is optional. Possible values are:

#### **READWrite**

Specifies that users and system processes (like migration) can read from and write to files stored on the volume.

If the volume being updated is an empty scratch volume that had an access mode of offsite, the volume is deleted from the ADSM database.



## UPDATE VOLUME

### READOnly

Specifies that users and system processes (like migration) can only read files stored on the volume.

If the volume being updated is an empty scratch volume that had an access mode of *offsite*, the volume is deleted from the ADSM database.

### UNAVailable

Specifies that neither users nor system processes (like migration) can access files stored on the volume.

If you update a random access volume to *unavailable*, you cannot vary the volume online.

If you update a sequential access volume to *unavailable*, ADSM does not attempt to mount the volume.

If the volume being updated is an empty scratch volume that had an access mode of *offsite*, the volume is deleted from the ADSM database.

### DESTROYed

Specifies that a primary storage pool volume has been permanently damaged. Neither users nor system processes (like migration) can access files stored on the volume.

Only volumes in primary storage pools can be updated to *destroyed*.

If you update a random access volume to *destroyed*, you cannot vary the volume online.

If you update a sequential access volume to *destroyed*, ADSM does not attempt to mount the volume.

This access mode is used to indicate an entire volume that should be restored by using the RESTORE STGPOOL command. After all files on a destroyed volume have been restored, the volume is automatically deleted from the database.

If a volume contains no files and the UPDATE VOLUME command is used to change the access mode to *destroyed*, the volume is deleted from the database.

### OFFsite

Specifies that a copy storage pool volume is at an offsite location from which it cannot be mounted. Only volumes in copy storage pools can be updated to *offsite*.

Use this mode to help you track volumes that you move to offsite locations. ADSM treats offsite volumes differently than other volumes:

- Mount requests are not generated for volumes located offsite
- Data can be reclaimed or moved from offsite volumes by retrieving files from other storage pools
- Empty, offsite scratch volumes are not deleted from the copy storage pool

## UPDATE VOLUME

### **Location**=*location*

Specifies the location of the volume. This parameter is optional. It can be specified only for volumes in sequential-access storage pools. The maximum length of the location is 255 characters. The location must be enclosed in quotation marks if it contains any blank characters. To remove a previously defined description, specify the null string ("").

**Note:** If you specify values for both the ACCESS and LOCATION parameters but the access mode cannot be updated for a particular volume, the location attribute is also not updated for that volume. For example, if you specify ACCESS=OFFSITE and LOCATION for primary storage pool volumes, neither the access nor location values are updated because these volumes cannot be given an access mode of OFFSITE.

### **WHERESTGpool**=*poolname*

Specifies the name of the storage pool for volumes to be updated. This parameter can be used to restrict the update by storage pool. This parameter is optional. You can use a pattern-matching expression to specify this name. If you do not specify a storage pool name, volumes belonging to any storage pool are updated.

### **WHEREDEVclass**=*devclassname*

Specifies the name of the device class for volumes to be updated. This parameter can be used to restrict the update by device class. This parameter is optional. You can use a pattern-matching expression to specify this name. If you do not specify a device class name, volumes with any device class are updated.

### **WHEREACcess**=*accessmode*

Specifies the current access mode of volumes to be updated. This parameter can be used to restrict the update to volumes that currently have the specified access mode.

This parameter is optional. You can specify more than one access mode in a list by separating each mode with a comma, with no intervening spaces. If you do not specify a value for this parameter, the update is not restricted by the volumes' current access mode. Possible values are:

#### **READWrite**

Specifies that volumes with an access mode of READWRITE are updated.

#### **READOnly**

Specifies that volumes with an access mode of READONLY are updated.

#### **UNAVailable**

Specifies that volumes with an access mode of UNAVAILABLE are updated.

#### **OFFsite**

Specifies that volumes with an access mode of OFFSITE are updated.

## UPDATE VOLUME

### **DEStroyed**

Specifies that volumes with an access mode of DESTROYED are updated.

### **WHEREStatus=*status***

Specifies the status of volumes to be updated. This parameter can be used to restrict the update to volumes that have a specified status. This parameter is optional. You can specify more than one status in a list by separating each status with a comma, with no intervening spaces. If you do not specify a value for this parameter, the updated volumes are not restricted by volume status. Possible values are:

### **ONline**

Specifies that volumes with a status of ONLINE are updated.

### **OFFline**

Specifies that volumes with a status of OFFLINE are updated.

### **EMPTy**

Specifies that volumes with a status of EMPTY are updated.

### **PENding**

Specifies that volumes with a status of PENDING are updated. These are volumes from which all files have been deleted, but the time specified by the REUSEDELAY parameter has not elapsed.

### **FILLing**

Specifies that volumes with a status of FILLING are updated.

### **FULI**

Specifies that volumes with a status of FULL are updated.

### **Preview=*value***

Specifies whether you want to preview the update operation without actually performing it. This parameter is optional. The default value is NO. Possible values are:

### **No**

Specifies that volumes are actually updated.

### **Yes**

Specifies that you only want to preview the update operation. This option displays the volumes that will be updated if you actually perform the update operation.

## Examples

### Task 1

Update a tape volume named DSMT20 to make it unavailable to users and system processes (like migration).

**Command:** `update volume dsmt20 access=unavailable`

## UPDATE VOLUME

### Output:

```
ANR2207I Volume DSMT20 updated.
```

### Task 2

Update all empty, offsite volumes in the COPYSTG storage pool. Set the access mode to READWRITE and delete the location information for the updated volumes.

**Command:** update volume \* access=readwrite location="" wherestgpool=copystg  
whereaccess=offsite,unavailable wherestatus=empty

### Output:

```
ANR2207I Volume COVOL1 updated.  
ANR2207I Volume COVOL2 updated.  
ANR2207I Volume COVOL3 updated.
```

## Related Commands

Table 177 provides references to related commands.

Table 177. Commands Related to UPDATE VOLUME

Command	Description	Reference
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
QUERY VOLUME	Displays information about one or more storage pool volumes.	Page 450
VARY	Specifies whether a disk volume is available to the server for use.	Page 629

## VALIDATE POLICYSET

---

### VALIDATE POLICYSET (Verify a Policy Set)

Use the VALIDATE POLICYSET command to verify that a policy set is complete and valid prior to activating it. The validate command examines the management class and copy group definitions in a specified policy set and reports on conditions that need to be considered if the policy set is to be activated.

The VALIDATE POLICY SET command will fail if any of the following conditions exist:

- A default management class is not defined for the policy set.
- A copy group within the policy set specifies a copy storage pool as a destination.
- A management class specifies a copy pool as the destination for space-managed files.

The following conditions result in warning messages issued during VALIDATE POLICYSET command processing:

- The destinations specified in the policy set copy group definitions do not refer to defined storage pools.  
If a policy set is activated with copy group definitions that do not specify valid storage pool names, the client backup or archive operations fail.
- The active policy set in a policy domain contains management classes that are not defined in the policy set being validated.  
If the policy set is activated, the backup files bound to the deleted management class are rebound to the default management class in the new active policy set.
- The active policy set in the policy domain contains copy groups that are not defined in the policy set being activated.  
Files bound to the management classes with deleted copy group definitions are no longer archived or backed up.
- The default management class does not contain a backup or archive copy group in the policy set being activated. If this default management class is used files are not backed up or archived by default.
- A storage pool specified as a destination for files migrated from client nodes (MIGDESTINATION) is not defined.
- A management class specifies that a backup version must exist before a file can be migrated from a client node (MIGREQUIRESBKUP=YES), but the management class does not contain a backup copy group.

After you are satisfied with the results of validation processing, use the ACTIVATE POLICYSET command to permit the attributes of the validated policy set to be defined in your storage hierarchy.

## VALIDATE POLICYSET

### Privilege Class

To issue this command, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the policy set belongs.

### Syntax

```
▶▶—VALidate POLicyset—domainname—setname————▶▶
```

### Parameters

*domainname*

Specifies the name of the policy domain to which the policy set is assigned. This parameter is required.

*setname*

Specifies the name of the policy set to be verified. This parameter is required.

### Examples

#### Task 1

Validate the policy set VACATION located in the EMPLOYEE\_RECORDS policy domain.

**Command:** `validate policyset employee_records vacation`

**Output:**

```
ANR1515I Policy set VACATION validated in
domain EMPLOYEE_RECORDS (ready for activation).
```

#### Task 2

Validate the policy set SET1 located in the PROG1 policy domain.

**Command:** `validate policyset prog1 set1`

**Output:**

```
ANR1553W DEFAULT Management class TOM in policy set SET1
PROG1 does not have a BACKUP copygroup: files will not be
backed up by default if this set is activated.
ANR1554W DEFAULT Management class TOM in policy set SET1
PROG1 does not have an ARCHIVE copygroup: files will not be
archived by default if this set is activated.
ANR1515I Policy set TOM validated in domain TOM (ready for
activation).
```

## VALIDATE POLICYSET

### Task 3

Validate the policy set STANDARD located in the STANDARD policy domain.

**Command:** `validate policyset standard standard`

**Output:**

**Note:** The following output is displayed when an invalid space management destination is specified.

```
ANR1557W The space management migration destination
in management class MCLASS2
does not refer to a defined storage pool: MCLASS2.
If this pool does not exist
when policy set STANDARD is activated, clients will fail
when using this management class to migrate space-managed
files to the server.
ANR1515I Policy set STANDARD validated in domain
STANDARD (ready for activation).
```

## VALIDATE POLICYSET

### Related Commands

Table 178 provides references to related commands.

*Table 178. Commands Related to VALIDATE POLICYSET*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
ACTIVATE POLICYSET	Validates and activates a policy set within a policy domain.	Page 46
COPY POLICYSET	Creates a copy of an existing policy set.	Page 94
DEFINE COPYGROUP	Defines a copy group for backup or archive processing within a specified management class.	Page 103
DEFINE MGMTCLASS	Defines a management class within the policy domain and policy set specified.	Page 157
DELETE POLICYSET	Deletes a policy set, including its management classes and copy groups, from a policy domain.	Page 226
QUERY POLICYSET	Displays information about one or more policy sets.	Page 415
UPDATE COPYGROUP	Changes one or more attributes of a copy group.	Page 539
UPDATE POLICYSET	Changes the description of a policy set.	Page 588



## VARY

---

### VARY (Bring a Random Access Volume Online or Offline)

Use the VARY command to make a random access volume available or unavailable to the server for use as a database, recovery log, or storage pool volume. This command is only valid for volumes that reside on random access devices. Examples of when you would use this command are during maintenance or corrective action of a random access volume.

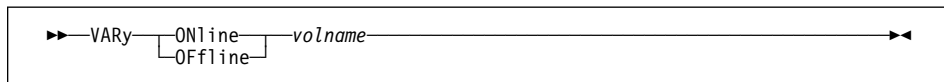
You cannot vary a random access volume online that is defined as *unavailable*.

Varying a database or recovery log volume online can start a background process when ADSM synchronizes the volume. Use the CANCEL PROCESS command to cancel this process. To display information on background processes, use the QUERY PROCESS command.

### Privilege Class

To issue this command, you must have system privilege or operator privilege.

### Syntax



### Parameters

#### ONline

Specifies that the random access volume can be used by the server as a database, recovery log, or storage pool volume.

#### OFFline

Specifies that the random access volume *cannot* be used by the server as a database, recovery log, or storage pool volume.

#### volname

Specifies the volume identifier. This parameter is required.

Any volume name used by ADSM cannot contain any embedded blanks or equal signs.

### Examples

#### Task 1

Make c:\admserv\stgvd.1 available to the server for use as a database volume.

**Command:** vary online c:\admserv\stgvd.1

## VARY

### Output:

```
ANR2210I Vary on-line initiated for disk volume
c:\admserv\stgvd.1.
ANR1305I Disk volume c:\admserv\stgvd.1 varied on-line.
```

### Task 2

Make c:\admserv\stgvd.1 unavailable to the server for use as a database volume.

**Command:** vary offline c:\admserv\stgvd.1

### Output:

```
ANR2211I Vary off-line initiated for disk
volume c:\admserv\stgvd.1.
ANR1307I Disk volume c:\admserv\stgvd.1 varied
off-line.
```

## Related Commands

Table 179 provides references to related commands.

Table 179 (Page 1 of 2). Commands Related to VARY

Command	Description	Reference
CANCEL PROCESS	Cancels a background server process.	Page 74
DEFINE DBVOLUME	Assigns a volume to be used for database storage.	Page 118
DEFINE LOGVOLUME	Defines a disk volume to the recovery log.	Page 155
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.	Page 193
DELETE DBVOLUME	Deletes a volume from the database.	Page 204
DELETE LOGVOLUME	Deletes a volume from the recovery log.	Page 222
DELETE VOLUME	Deletes a volume from a storage pool.	Page 240
QUERY DBVOLUME	Displays information about one or more database volumes.	Page 347

## VARY

*Table 179 (Page 2 of 2). Commands Related to VARY*

<b>Command</b>	<b>Description</b>	<b>Reference</b>
QUERY LOGVOLUME	Displays information about one or more log volumes.	Page 394
QUERY PROCESS	Displays information about one or all background server processes.	Page 419
QUERY VOLUME	Displays information about one or more storage pool volumes.	Page 450

**VARY**

---

## Appendix A. Miscellaneous Utilities

The following utilities are available:

- DSMFMT** Use to pre-allocate space for database, recovery log and storage pool volumes.
- DSMLABEL** Use to write labels to removable media storage volumes in preparation for using them with the ADSM server.
- GETIPXAD** Use to display the IPX/SPX address of the OS/2 system on which ADSM for OS/2 is running. The address is needed for configuring the IPX/SPX protocol on your clients. This command is run on the system on which you will run the server.
- Note:** IPX/SPX must be active for this command to obtain the IPX/SPX address of your system.
- UPGRADEDDB** Use to upgrade an ADSM Version 1 server to an ADSM Version 2 server.

## DSMFMT

---

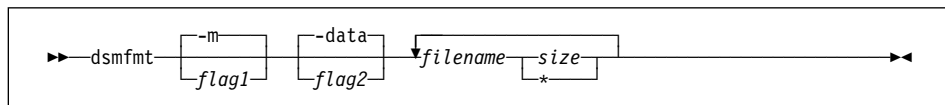
### DSMFMT (Formatting Files for the Server)

The DSMFMT utility is used to allocate the file space needed for database, recovery log, and storage pool volumes. This utility is delivered as part of the ADSM server; it is not a server command. The parameters are not case-sensitive and do not have abbreviations.

**Note:** For performance reasons, allocate database, recovery log, and storage pool volumes:

- In an HPFS file system, not an FAT file system.
- On disk drives that reside on the ADSM server machine, not on remotely mounted file systems.

### Syntax



### Parameters

#### *flag1*

Specifies the size in either kilobytes or megabytes. The default size is megabytes. This option is positional and is not case-sensitive. Possible values are:

**-k**

Specifies kilobytes are to be used.

**-m**

Specifies megabytes are to be used; this is the default value.

#### *flag2*

Specifies the intended use of the files to be formatted. The default is `-data`. This option is positional and is not case-sensitive. Possible values are:

**-db**

Specifies the file is for a database volume.

**-log**

Specifies the file is for a recovery log volume.

**-data**

Specifies the file is for a storage volume in data storage.

#### *filename*

Specifies the name of the file to be formatted. This filename must not already exist.

#### *size*

Specifies the size of the file to be formatted, expressed as a positive integer or an asterisk (\*).

## DSMFMT

- For database and recovery log files, the allocated file size is always a multiple of 4MB (plus an additional 1MB for overhead). The minimum file sizes are 9MB for the recovery log and 5MB for the database. For data storage files, the minimum size is 1MB.
- If you use an asterisk (\*) for the file size, the size will equal the remaining space minus 32KB. The minimum requirements for size must still be met.
- The maximum size is 2GB minus 1 byte for database, recovery log, and storage files.

### Usage Notes

- The DSMFMT utility does not destroy or modify an existing file.
- If a file does not format, DSMFMT issues an error message for that file and then continues to format the next file on the list.

### Examples

An example of formatting two recovery log volumes each having 9 megabytes of storage follows:

**Command:** `dsmfmt -m -log logvol1 9 logvol2 9`

```
ADSTAR Distributed Storage Manager/2
OS/2 ADSM Server DSMFMT Extent/Volume Formatting Program

Allocated space for logvol1: 9437184 bytes
Allocated space for logvol2: 9437184 bytes
```

An example of formatting a database volume having 5 megabytes of storage follows:

**Command:** `dsmfmt -m -db dbvol1 5`

```
ADSTAR Distributed Storage Manager/2
OS/2 ADSM Server DSMFMT Extent/Volume Formatting Program

Allocated space for dbvol1: 5242880 bytes
```

An example of formatting storage volumes backvol and archvol having 21MB and 8MB of storage, respectively, follows:

**Command:** `dsmfmt -m -data backvol 21 archvol 8`

## DSMFMT

```
ADSTAR Distributed Storage Manager/2  
OS/2 ADSM Server DSMFMT Extent/Volume Formatting Program
```

```
Allocated space for backvol: 22020096 bytes  
Allocated space for archvol: 8388608 bytes
```



## DSMLABEL

### DSMLABEL (Labeling Volumes)

Use the DSMLABEL utility to write labels to storage volumes in preparation for using them with the ADSM server. The utility works with all tape and optical drives, whether they are stand-alone units or are in libraries. The parameters are not case-sensitive and do not have abbreviations.

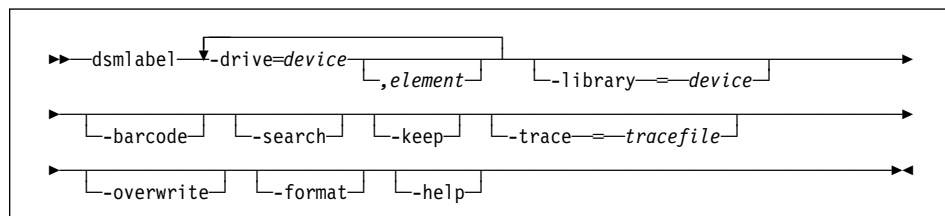
**Attention:** Prior to relabeling a volume existing data must be removed. Before relabeling a volume that has been in use by ADSM, you can either move the data via the MOVE DATA command or delete the contents of the volume via the DELETE VOLUME command with the DISCARDATA parameter equal to YES.

It is possible to encounter an error when DSMLABEL is processing and the ADSM server is simultaneously accessing drives that DSMLABEL is using. To avoid this error perform either of the following:

- Be sure that the ADSM server is not running during DSMLABEL processing.
- Temporarily delete the drive from ADSM that is being used by DSMLABEL. If the drive deletion is for a long period of time, you should decrease the mountlimit for the device class pertaining to this deleted drive, see the UPDATE DEVCLASS command.

REMOVABLEFILE volumes are labelled with either the device manufacturer's utilities or the operating system utilities (the operating system utilities are the label command for the command line and the disk administrator program for the GUI interface, such as the label command).

### Syntax



### Parameters

#### **-drive=device**

Specifies a drive to use. You must specify at least one drive. If the drive is within a multidrive SCSI library, the drive's element address must also be provided. The maximum number of drives you can specify is eight.

#### **element**

This parameter is optional and is used only if you specify a drive within a multidrive SCSI library. It must be preceded by a comma, with no spaces between the device name and the element address. For element addresses, see *ADSM Administrator's Guide*.

## DSMLABEL

### **-library=*device***

Specifies a library to use for volume labeling. This parameter is optional. If it is omitted, you will be prompted to manually load and unload volumes. If specified, it indicates that the drives reside in a library and can be accessed using a robotic device. Only one library can be specified.

### **-barcode**

Specifies to utilize the bar code reader in the SCSI device. This parameter is optional. The first six characters of the bar code are used as the volume name instead of prompting you for a volume name.

### **-search**

Causes the labeling utility program to search the specified library for all usable volumes and to write a label to each one found. This parameter is optional; it has no effect unless the `-library` parameter is specified. If it is omitted, and the `-library` parameter is specified, the operator is prompted to manually load and remove volumes from the library's entry/exit port, if available. If the library is not equipped with an entry/exit port, the operator is prompted to load and unload volumes from the drives in the library.

### **-keep**

This option causes the program to keep volumes inside the library after they are labeled. If it is omitted, volumes are returned to the library's entry/exit port after being successfully labeled. If the library is not equipped with an entry/exit port, the operator is prompted to remove volumes from the drive. This option has no effect unless the `-library` parameter is specified.

### **-trace=*tracefile***

Creates a trace file for use by IBM service personnel. This parameter is optional and should be used only if there is a problem with the program.

### **-overwrite**

Allows the program to overwrite an existing label. This parameter is optional. If omitted, any volume with a preexisting label is rejected and the next unlabeled volume is requested.

### **-format**

Indicates that you want the media to be formatted. The only device type for which this option has an effect is a 5 1/4-inch optical drive.

### **-help**

Displays the online help information. If you enter `dsmlabel` without any parameters, you will automatically receive the help screen information.

## Examples

```
dsmlabel -drive=tape05$
dsmlabel -drive=tape05$ -library=auto04$ -keep -overwrite
dsmlabel -drive=tape05$ -library=auto04$ -search
dsmlabel -drive=h: -format
```

---

## GETIPXAD (Get the IPX Address)

Use the GETIPXAD command to display the IPX/SPX address of an OS/2 system on which ADSM for OS/2 is running. This address is needed for configuring the IPX/SPX protocol on an OS/2 server. The command is run on the system on which you will run the server.

**Note:** IPX/SPX must be active for this command to obtain the IPX/SPX address of your system.

### Syntax

```
▶▶ GETIPXAD socketnumber ▶▶
```

### Examples

An example showing the socket number set to 8522 follows:

**Command:** `getipxad`

The output is an IPX/SPX address similar to the following:

```
000000020004ac33dd21
```

where:

**00000002** is the network number

**0004ac33dd21** is the node address

The information displayed below should be entered in your client options file to allow the client to communicate with the server:

```
COMMMETHOD IPXSPX
IPXServeraddress 000000020004ac33dd21
IPXSocket 8522
```

For example, the default socket number of 8522 can be changed within the server options file by entering:

```
IPXSocket 8510
```

## UPGRADEDDB

---

### UPGRADEDDB (Upgrading from Version 1 to Version 2)

The UPGRADEDDB utility specifies whether the database should be upgraded from an ADSM Version 1 server to an ADSM Version 2 server. This utility should be specified only if you have previously been running an ADSM Version 1 server and it should be specified only the first time you start the Version 2 server.

Before specifying this utility, you should ensure that you have backed up existing copies of your database and recovery log files. Once the upgrade has occurred, you will not be able to revert to a Version 1 server.

### Syntax

▶▶—UPGRADEDDB—————▶▶

### Parameters

None.

### Examples

None.

---

## Appendix B. Recovery Log Extension Command

The following command is used to extend the size of the recovery log in situations that the server is halted due to an out-of-space (over committed) condition. The command is:

### **DSMSERV EXTEND LOG**

Use to extend the size of the recovery log.

## DSMSERV EXTEND LOG

---

### DSMSERV EXTEND LOG (Emergency Log Extension)

Use the DSMSERV EXTEND LOG command to extend the size of the recovery log when you require additional log space to successfully restart the server.

When using the ADSM server with the logmode set to rollforward, it is possible for the server to run out of recovery log space before database operations can complete successfully. This out-of-space (over-committed) condition can occur when the recovery log is too small or a database backup trigger is set so that the database backup log full percentage is too high. In either of these scenarios, you may either need to halt the server without sufficient recovery log space available for a successful server restart or the server will issue the following message and then halt:

```
ANR9999D LOGSEG(415) Log Space Has Been Over committed.
```

In these situations additional recovery log space is required to restart the server. By using DSMSERV EXTEND LOG, additional space is allocated allowing for a successful server restart. Once the server is running, a database backup can occur, thereby freeing additional recovery log space.

### Syntax

```
▶▶—DSMSERV EXTEND LOG—volname—size—————▶▶
```

### Parameters

*volname*

Specifies the name of a new, formatted volume available for use as the recovery log extension. This parameter is required. Use the DSMFMT utility to format the new volume.

*size*

Specifies the size, in megabytes, of the recovery log extension. This parameter is required. The specified size should be a multiple of 4 and at least 1 MB less than the actual volume size of the volume specified in the *volname* parameter. For example, if the volume specified in the *volname* parameter is 5MB, the entered value for the *size* parameter is 4.

### Usage Notes

- The DSMFMT utility formats new volumes to use with the DSMSERV EXTEND LOG command.
- The server can be restarted after successfully extending the recovery log. At this time, the size of the recovery log, the database backup trigger full percentage, or both need to be adjusted to allow for successful future database backups.

## DSMSERV EXTEND LOG

### Examples

#### Task

Assume that the server has used all available recovery log space; therefore, a database backup operation cannot successfully complete. The server has halted and during server restart, insufficient recovery log space is available for the restart. Using the DSMFMT utility, a newly formatted 21MB volume is made available for a recovery log extension. The volume name is reclog.new. The task is to extend the recovery log using this newly formatted volume. Once the recovery log extension completes, the server can be restarted into a production mode.

**Command:** dsmserv extend log reclog.new 20

#### Output:

```
ANR7800I DSMSERV generated at 11:15:01 on Jun  3 1996.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0307I Recovery log extend in progress; 4 megabytes of 20 formatted.
ANR0307I Recovery log extend in progress; 8 megabytes of 20 formatted.
ANR0307I Recovery log extend in progress; 12 megabytes of 20 formatted.
ANR0307I Recovery log extend in progress; 16 megabytes of 20 formatted.
ANR0307I Recovery log extend in progress; 20 megabytes of 20 formatted.
ANR2268I Recovery log assigned capacity has been extended.
ANR7835I The ADSM thread 1 (tid 2) terminated in response to server
shutdown.
ANR7835I The ADSM thread 4 (tid 5) terminated in response to server
shutdown.
ANR7835I The ADSM thread 5 (tid 6) terminated in response to server
shutdown.
ANR0991I ADSM server shutdown complete.
```

## DSMSERV EXTEND LOG



---

## Appendix C. Database Recovery Commands

The following commands are used to recover your server database if a catastrophic error occurs, and you have a backup copy of the database available. In alphabetical order, the commands are:

### **DSMSERV DISPLAY DBBACKUPVOLUME**

Use to display information about database backup volumes when volume history information is not available.

### **DSMSERV DISPLAY DBVOLUMES**

Use to display information about all database volumes known to the server.

### **DSMSERV DISPLAY LOGVOLUMES**

Use to display information about all recovery log volumes known to the server.

### **DSMSERV RESTORE DB**

Use to restore a database volume or an entire database, either to its most current state or to a specific point in time. The command uses backup versions of the database created explicitly via the BACKUP DB command or automatically via the database backup trigger.

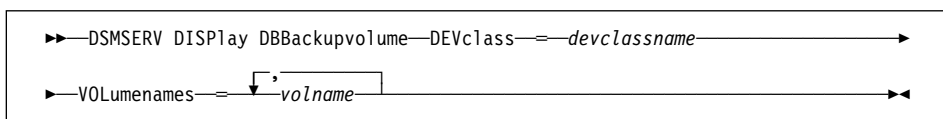
## DSMSERV DISPLAY DBBACKUPVOLUME

---

### DSMSERV DISPLAY DBBACKUPVOLUME (Display Database Backup Volume Information)

Use the DSMSERV DISPLAY DBBACKUPVOLUME command to display information about database backup volumes when volume history information is not available.

#### Syntax



#### Parameters

##### **DEVclass=devclassname**

Specifies the name of the sequential access device class to use. The server options file that is created during installation includes a DEVCONFIG option that specifies an external file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. Any device class you specify must be defined in an external file specified with the DEVCONFIG option. This parameter is required.

If the device configuration file does not exist or has been lost, you can manually create one. See *ADSM Administrator's Guide*.

##### **VOLUmenames=volname**

Specifies the database backup volumes for which you want to display information. This parameter is required. You can specify more than one database backup volume by separating each volume name with a comma, with no intervening spaces.

#### Examples

##### Task

Display information about the full database backup volume, BACKUPFILE and the incremental database backup volume, INCREMENTAL1.

**Command:** dsmserve display dbbackupvolume devclass=file  
volumenames=backupfile,incremental1

## DSMSERV DISPLAY DBBACKUPVOLUME

### Output:

```
ADSTAR Distributed Storage Manager for OS/2
Version 2, Release 1, Level 0.8/0.8

ANR7800I DSMSERV generated at 11:15:01 on Jun  3 1996.
ANR8324I FILE volume D:\ADSMV2\BACKUP is expected to be mounted (R/O).
ANR8324I FILE volume D:\ADSMV2\INCREMENTAL is expected
to be mounted (R/O).
ANR8340I FILE volume D:\ADSMV2\BACKUP mounted.
ANR1363I Input volume D:\ADSMV2\BACKUP opened (sequence number 1).
ANR1364I Input volume D:\ADSMV2\BACKUP closed.

        Date/Time: 06/17/1996 12:35:48
        Volume Type: BACKUPFULL
        Backup Series: 1
        Backup Operation: 0
        Volume Seq: 1
        Device Class: FILE
        Volume Name: D:\ADSMV2\BACKUP

ANR8340I FILE volume D:\ADSMV2\INCREMENTAL mounted.
ANR1363I Input volume D:\ADSMV2\INCREMENTAL opened (sequence number 1).
ANR1364I Input volume D:\ADSMV2\INCREMENTAL closed.

        Date/Time: 06/17/1996 12:37:09
        Volume Type: BACKUPINCR
        Backup Series: 1
        Backup Operation: 1
        Volume Seq: 1
        Device Class: FILE
        Volume Name: D:\ADSMV2\INCREMENTAL
```

where:

#### **Date/Time**

The date and time that backup data was recorded on the volume.

#### **Volume Type**

The type of backup data on the volume, that is, whether the backup recorded on the volume is a full or an incremental backup.

#### **Backup Series**

The identifying number of the backup series of which this volume is a part. Each backup series consists of a full backup and all incremental backups that apply to that full backup, up to the next full backup of the database.

#### **Backup Operation**

The operation number of this backup volume, within the backup series. The full backup within a backup series is operation 0. The first incremental backup for that full backup is operation 1, the second incremental backup is operation 2, and so on.

## **DSMSERV DISPLAY DBBACKUPVOLUME**

### **Volume Seq**

The sequence, or position, of the volume within the backup series. Volume sequence 1 identifies the first volume used for the first operation (a full backup), and so on. For example, if the full backup occupies three volumes, these volumes are identified as volume sequence 1, 2, and 3, respectively. The first volume of the next operation (the first incremental backup) is then volume sequence 4.

### **Device Class**

The name of the device class that this volume is assigned to.

### **Volume Name**

The name of the volume.

## DSMSERV DISPLAY DBVOLUMES

---

### DSMSERV DISPLAY DBVOLUMES (Display Database Volume Information)

Use the DSMSERV DISPLAY DBVOLUMES command to display information about all database volumes known to the ADSM server.

#### Syntax

```
▶—DSMSERV DISPlay DBVoluMes—▶▶
```

#### Parameters

None.

#### Examples

##### Task

Display information about all database volumes.

**Command:** dsmserv display dbvolumes

##### Output:

```
Volume Name (Copy 1): D:\DSMSTG\DB0
    Copy Status: Sync'd
Volume Name (Copy 2):
    Copy Status: Undefined
Volume Name (Copy 3):
    Copy Status: Undefined
Available Space (MB): 4
Allocated Space (MB): 4
    Free Space (MB): 0
```

where:

##### Volume Name (Copy *n*)

Specifies the name of the database volume. The output displays the status of the three possible volume copies (Copy 1, Copy 2, and Copy 3).

##### Copy Status

Specifies the status of the database volume. Possible values are:

##### Offline

The volume is varied offline.

##### Stale

The volume is not synchronized with the other volumes in the database and cannot be used for storage.

##### Sync'd

The volume is synchronized with the other volumes in the database and can be used for storage.

## DSMSERV DISPLAY DBVOLUMES

### **Undefined**

No volume has been defined.

### **Available Space (MB)**

Specifies the total amount of space that is available, both allocated and free, for the database volume, in megabytes.

### **Allocated Space**

Specifies the amount of space allocated to the database volume, in megabytes.

### **Free Space**

Specifies the amount of unassigned space for this mirror set that can be allocated for use, in megabytes.

## DSMSERV DISPLAY LOGVOLUMES

### DSMSERV DISPLAY LOGVOLUMES (Display Recovery Log Volume Information)

Use the DSMSERV DISPLAY LOGVOLUMES command to display information about all recovery log volumes known to the ADSM server.

#### Syntax

```
▶—DSMSERV DISPlay LOGVoLumes—◀◀
```

#### Parameters

None.

#### Examples

##### Task

Display information about all recovery log volumes.

**Command:** dsmserv display logvolumes

##### Output:

```
Volume Name (Copy 1): D:\DSMSTG\LOG0
    Copy Status: Sync'd
Volume Name (Copy 2):
    Copy Status: Undefined
Volume Name (Copy 3):
    Copy Status: Undefined
Available Space (MB): 8
Allocated Space (MB): 8
    Free Space (MB): 0

Volume Name (Copy 1): D:\ADSMV2\RECL0G.NEW
    Copy Status: Sync'd
Volume Name (Copy 2):
    Copy Status: Undefined
Volume Name (Copy 3):
    Copy Status: Undefined
Available Space (MB): 20
Allocated Space (MB): 20
    Free Space (MB): 0
```

where:

##### Volume Name (Copy *n*)

Specifies the name of the recovery log volume. The output displays status of the three possible volume copies (Copy 1, Copy 2, and Copy 3).

##### Copy Status

Specifies the status of the recovery log volume. Possible values are:

## DSMSERV DISPLAY LOGVOLUMES

**Offline**

The volume is varied offline.

**Stale**

The volume is not synchronized with the other volumes in the recovery log and cannot be used for storage.

**Sync'd**

The volume is synchronized with the other volumes in the recovery log and can be used for storage.

**Undefined**

No volume has been defined.

**Available Space (MB)**

Specifies the total amount of space that is available, both allocated and free, for the recovery log, in megabytes.

**Allocated Space**

Specifies the amount of space allocated for the recovery log, in megabytes.

**Free Space**

Specifies the amount of unassigned space in the recovery log that can be allocated for use, in megabytes.



## DSMSERV RESTORE DB

---

### DSMSERV RESTORE DB (Restore the Database)

Use the DSMSERV RESTORE DB command to:

- Restore an ADSM database to its most current state
- Restore a single ADSM database volume to its most current state
- Restore an ADSM database to a specific point in time when the volume history file is available
- Restore an ADSM database to a specific point in time when the volume history file is not available

The DSMSERV RESTORE DB command uses backup versions of the database you create using the BACKUP DB command or backup versions created automatically by ADSM based on the settings you specify with the DEFINE DBBACKUPTRIGGER command. You cannot use the DSMSERV RESTORE DB command to load an offline salvage dump created with the DSMSERV DUMP DB command.

To restore an ADSM database, you can use one of the following:

#### **Roll-forward recovery**

Restores the database or a single database backup volume to its most current state by first loading the most recent database backup series (the last full backup and any incremental backups that apply to that full backup), and then using recovery log records to apply any changes made to the database since the last backup was created

#### **Point-in-time recovery**

Restores the database to the point in time when a specific backup version was created

After you perform a point-in-time recovery, you must use the AUDIT VOLUME command to audit all DISK volumes and to check for and resolve any inconsistencies between database information and storage pool volumes. If you saved a copy of the volume history file before restoring the database, examine that file to find out about sequential access storage pool volumes that were deleted or reused since the point in time to which the database was restored.

The syntax and parameters for the DSMSERV RESTORE DB command vary depending on the type of database recovery you want to perform. The types are:

- Restoring a database to its most current state
- Restoring a single database volume to its most current state
- Restoring a database to a specific point in time—volume history file available
- Restoring a database to a specific point in time—volume history file unavailable

### **Syntax for Restoring a Database to Its Most Current State**

You can perform a roll-forward recovery for an entire database, using the syntax that follows, if the following are true:

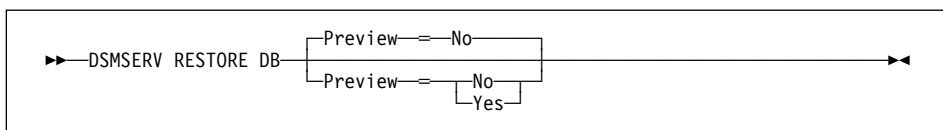
- The log mode was set to ROLLFORWARD continuously from the time that the last backup series was created, until the time that the database was damaged or lost.

## DSMSERV RESTORE DB

- The recovery log is available and intact.
- A volume history file is available and intact.

When you enter the DSMSERV RESTORE DB command with no parameters, ADSM restores the database to its most current state as follows:

1. Reads the recovery log to determine which volumes contain the most recent database backup series.
2. Requests volume mounts to load the most recent backup series (the last full backup and any incremental backups that apply to that full backup). The requests for mounts and the loading continue until the entire backup series is loaded.
3. Uses the recovery log to update the database to its most current state.



### Parameters for Restoring a Database to Its Most Current State

#### **Preview=value**

Specifies whether you want ADSM to display a list of the volumes needed to restore the database to its most current state, without performing the restore operation. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that you want to restore the database.

#### **Yes**

Specifies that you only want to display the list of database backup volumes, without actually restoring the database.

### Syntax for Restoring a Single Database Volume to Its Most Current State

You can perform a roll-forward recovery for a single database volume, using the syntax that follows, if both of the following are true:

- The log mode was set to ROLLFORWARD continuously from the time that the last backup series was created, until the time that the database volume was damaged or lost.
- The recovery log is available and intact.

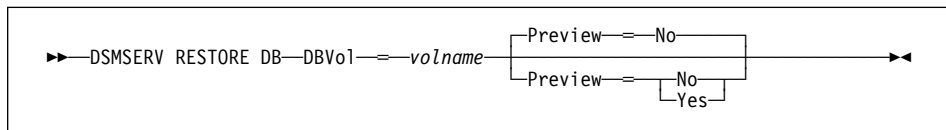
When you use the DSMSERV RESTORE DB command to restore a single database volume, ADSM performs the same steps as those described in “Syntax for Restoring a Database to Its Most Current State” on page 653, except that it restores only the pages that correspond to the database volume specified.

When ADSM restores a single database volume that is mirrored, only one copy of the mirrored volumes is restored. You must use the VARY command with the ONLINE

## DSMSERV RESTORE DB

parameter to make the mirrored volumes available again. ADSM then synchronizes the mirrored volumes with the restored volume.

The DBVOL option applies to the current database configuration only.



### Parameters for Restoring a Single Database Volume to Its Most Current State

#### **DBVol=volname**

Specifies the name of the database volume to restore. This parameter is required.

If the database volume is mirrored, you can specify the name of any volume that contains a mirrored copy of the volume you want to restore. Use the `DSMSERV DISPLAY DBVOLUMES` command to find the volume names.

#### **Preview=value**

Specifies whether you want ADSM to display a list of the database backup volumes needed to restore the database volume you specify, without performing the restore. This parameter is optional. The default value is NO. Possible values are:

##### **No**

Specifies that you want to restore the database volume.

##### **Yes**

Specifies that you only want to display the list of database backup volumes needed, without actually restoring the database.

### Syntax for Restoring a Database to a Specific Point in Time—History File Available

If a backup copy of volume history information is available, you can restore a database to a specific point in time using the syntax defined below. The server options file that is created during installation includes `VOLUMEHISTORY` and `DEVCONFIG` options that specify default files named `volhist.out` and `devcnfg.out`, respectively. The server program must be able to open and read the files successfully when running the `DSMSERV RESTORE DB` command.

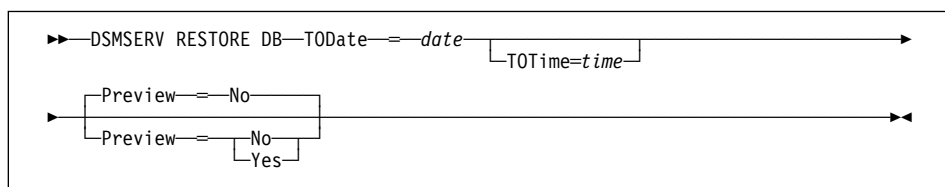
If the volume history file is not available, and you want to perform a point-in-time recovery, you can use the recovery method described in “Syntax for Restoring a Database to a Specific Point in Time—History File Unavailable” on page 657.

If a device configuration file is not available, you can manually create one. See *ADSM Administrator's Guide*.

## DSMSERV RESTORE DB

ADSM restores the database as follows:

1. Uses the volume history file to locate the database backup volumes needed
2. Requests mounts and loads data from database backup volumes as required to restore the database volume to the specified time



### Parameters for Restoring a Database to a Specific Point in Time—History File Available

#### **TODate=*date***

Specifies the date to which to restore the database. This parameter is required. Possible values are:

##### *MM/DD/YYYY*

Specifies that you want to restore a database using the most recent backup series that was created prior to a specific date.

##### **TODAY**

Specifies that you want to restore a database using the most recent backup series that was created prior to the current date.

##### **TODAY**-*numdays* or -*numdays*

Specifies that you want to restore a database using the most recent backup series that was created prior to a specific number of days before the current date.

For example, to restore a database using the most recent backup series that was created prior to one week ago, you can enter either of the following:

```
dsmserv restore db todate=today-7
dsmserv restore db todate=-7
```

#### **TOTime=*time***

Specifies the time of day to which to restore the database. This parameter is optional. The default is the end of the day (23:59:59). Possible values are:

##### *HH:MM:SS*

Specifies that you want to restore the database using the last backup series created on or before a specific time on the specified date.

##### **NOW**

Specifies that you want to restore the database using a backup series created on or before the current time on the specified date.

## DSMSERV RESTORE DB

For example, if you issue the DSMSERV RESTORE DB command at 9:00 with TOTIME=NOW, ADSM restores the database using the last backup series created on or prior to 9:00 on the specified date.

### **NOW**–*numhours:numminutes* or –*numhours:numminutes*

Specifies that you want to restore the database using a backup series created on or before the current time minus a specified number of hours (and optionally, minutes) on the specified date.

For example, if you issue the DSMSERV RESTORE DB command at 9:00 with TOTIME=NOW–3:30 or TOTIME=–3:30, ADSM restores the database using the last backup series created on or prior to 5:30 on the specified date.

The following are sample commands to perform that task:

```
dsmserv restore db todate=today-7 totime=now-3:30
dsmserv restore db todate=-7 totime=-3:30
```

### **Preview**=*value*

Specifies whether you want ADSM to display a list of the volumes needed to restore the database to the date and time you specify, without actually performing the restore operation. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that you want to perform the restore operation.

#### **Yes**

Specifies that you only want ADSM to display the list of volumes needed to restore the database to the date and time you specify, without actually performing the restore operation.

## Syntax for Restoring a Database to a Specific Point in Time—History File Unavailable

If the volume history file is unavailable, you can use one or more DSMSERV RESTORE DB commands with the syntax shown below to restore an ADSM database to a specific point in time.

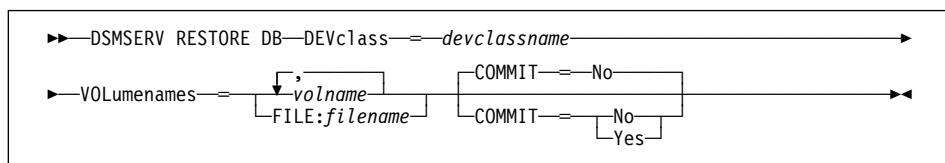
For example, if you need to load a full backup *and* one or more incremental backups, you can issue a DSMSERV RESTORE DB command to restore the full backup, then issue an additional DSMSERV RESTORE DB command for each incremental backup.

When you use multiple DSMSERV RESTORE DB commands, you must specify COMMIT=NO for each command except the last one you issue. For the last DSMSERV RESTORE DB command, you must specify COMMIT=YES to place the database in a consistent and usable state.

If you issue a DSMSERV RESTORE DB command with a COMMIT=NO parameter, and you decide not to load additional incremental backups, you must reissue the last DSMSERV RESTORE DB command you issued and specify COMMIT=YES.

## DSMSERV RESTORE DB

If you issue a DSMSERV RESTORE DB command with a COMMIT=YES parameter, and you decide that you need to load one or more additional incremental backups, you must reissue all of the DSMSERV RESTORE DB commands in order, starting with the restore of the full backup and specifying COMMIT=NO for each one. Then, you can enter one or more additional DSMSERV RESTORE DB commands, and specify COMMIT=YES for the last one.



### Parameters for Restoring a Database to a Specific Point in Time—History File Unavailable

#### **DEVclass=devclassname**

Specifies the name of the sequential access device class to use. The server options file that is created during installation includes a DEVCONFIG option that specifies a default file named devcnfg.out. This parameter is required.

If the device configuration file has been lost, you can manually create one. See *ADSM Administrator's Guide*.

#### **VOLUmenames=volname**

Specifies the backup volumes to use to restore the database. This parameter is required. Possible values are:

##### *volname*

Specifies the names of one or more volumes to use to restore the database. You can specify more than one backup volume by separating each volume name with a comma, with no intervening spaces. You must list the volumes in the order in which the backup was stored.

For example, if you want to use the backup volumes TAPE01, TAPE02, and TAPE03, specify the following:

```
VOLUMENAMES=TAPE01,TAPE02,TAPE03
```

##### **FILE:filename**

Specifies the name of a file that contains a list of the volumes to use to restore the database. The file name can be any valid AIX file name. In the file, each volume name must be on a separate line. You must list the volumes in the order in which the backup was stored.

For example, if you want to use the backup volumes TAPE01, TAPE02, and TAPE03, create a file named VOLLIST that contains these lines:

```
TAPE01  
TAPE02  
TAPE03
```

You can then specify the volumes for the restore command as follows:

## DSMSERV RESTORE DB

VOLUMENAMES=FILE:VOLLIST

### **COMMIT**=*commitvalue*

Specifies whether this is the last restore command you intend to use to restore the database. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that you intend to issue one or more additional DSMSERV RESTORE DB commands that list additional volumes. The database remains in an inconsistent and unusable state until you issue a DSMSERV RESTORE DB command with a COMMIT=YES parameter.

Specify COMMIT=NO if you plan to load additional backup volumes that contain one or more incremental backups.

#### **Yes**

Specifies that this is the last restore command you intend to use to restore the database. With this command, you are specifying the last backup volumes you want to use for the restore operation.

Specify COMMIT=YES only when you want to restore the database to the end of the last volume specified, and you do not plan to restore any additional backup volumes that contain incremental backups.

## Examples

### Task 1

Preview the list of volumes required to restore the database to its most current state.

**Command:** dsmserv restore db preview=yes

#### **Output:**

```
ADSTAR Distributed Storage Manager for OS/2
Version 2, Release 1, Level 0.8/0.8

ANR7800I DSMSERV generated at 11:15:01 on Jun  3 1996.
ANR0200I Recovery log assigned capacity is 28 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR4620I Database backup series 3 operation 0 device class FILE.
ANR4622I Volume 1: D:\ADSMV2\35034649.DBB.
ANR4620I Database backup series 3 operation 1 device class FILE.
ANR4622I Volume 1: D:\ADSMV2\35034819.DBB.
```

### Task 2

Restore the database to its state on June 17, 1996 at 2:30 p.m.

**Command:** dsmserv restore db todate=06/17/1996 totime=14:30 preview=no

## DSMSERV RESTORE DB

### Output:

```
ADSTAR Distributed Storage Manager for OS/2
Version 2, Release 1, Level 0.8/0.8

ANR0200I Recovery log assigned capacity is 28 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
ANR4600I Processing volume history file volhist.
ANR8324I FILE volume D:\ADSMV2\35036143.DBB is expected to be
mounted (R/O).
ANR8324I FILE volume D:\ADSMV2\35036207.DBB is expected to be
mounted (R/O).
ANR4620I Database backup series 4 operation 0 device class FILE.
ANR4622I Volume 1: D:\ADSMV2\35036143.DBB.
ANR4620I Database backup series 4 operation 1 device class FILE.
ANR4622I Volume 1: D:\ADSMV2\35036207.DBB.
ANR4634I Starting point-in-time database restore to date 06/17/1996
14:30:07.
ANR8340I FILE volume D:\ADSMV2\35036143.DBB mounted.
ANR1363I Input volume D:\ADSMV2\35036143.DBB opened (sequence number 1).
ANR4646I Database capacity required for restore is 4 megabytes.
ANR4638I Restore of backup series 4 operation 0 in progress.
ANR0300I Recovery log format started; assigned capacity 28 megabytes.
ANR0301I Recovery log format in progress; 4 megabytes of 28.
ANR0301I Recovery log format in progress; 8 megabytes of 28.
ANR0301I Recovery log format in progress; 12 megabytes of 28.
ANR0301I Recovery log format in progress; 16 megabytes of 28.
ANR0301I Recovery log format in progress; 20 megabytes of 28.
ANR0301I Recovery log format in progress; 24 megabytes of 28.
ANR1364I Input volume D:\ADSMV2\35036143.DBB closed.
ANR4640I Restored 101 pages from backup series 4 operation 0.
ANR8340I FILE volume D:\ADSMV2\35036207.DBB mounted.
ANR1363I Input volume D:\ADSMV2\35036207.DBB opened (sequence number 1).
ANR4638I Restore of backup series 4 operation 1 in progress.
ANR0301I Recovery log format in progress; 28 megabytes of 28.
ANR0302I Recovery log formatting took 21106 milliseconds.
ANR4640I Restored 10 pages from backup series 4 operation 1.
ANR0303I Format rate: 339.6 pages/second.
ANR0304I Page service time: 2.9 ms.
ANR0305I Recovery log format complete.
ANR0306I Recovery log volume mount in progress.
ANR4641I Sequential media log redo pass in progress.
ANR4642I Sequential media log undo pass in progress.
ANR1364I Input volume D:\ADSMV2\35036207.DBB closed.
ANR4644I A full backup will be required for the next database backup
operation.
ANR4635I Point-in-time database restore complete, restore date
06/17/1996 14:30:07.
```



---

## Appendix D. Database Salvage Commands

The following commands are used to re-establish your server database if a catastrophic error occurs, and the database cannot be recovered effectively using the DSMSERV RESTORE DB command. These commands are presented in the order in which they must be executed, as follows:

### **DSMSERV DUMPDB**

Use to dump the damaged database onto a new database.

### **DSMSERV INSTALL**

Use to initialize the server's database and recovery log.

### **DSMSERV LOADDB**

Use to reload the database.

### **DSMSERV AUDITDB**

Use to ensure that the database is returned to a consistent state after it is reloaded.

The DSMSERV DUMPDB and DSMSERV LOADDB commands use the device configuration file. The server options file that is created during installation includes a DEVCONFIG option that specifies an external file named devcnfg.out. This file contains a copy of the device class, library, and drive definitions that exist in your database. You can also create the file by using the BACKUP DEVCONFIG command. If no device configuration file has been maintained or created, you must manually create one with an editor. For information on how to create a device configuration file manually, see *ADSM Administrator's Guide*.

## DSMSERV DUMPDB

### DSMSERV DUMPDB (Dump the Database)

Use the DSMSERV DUMPDB command as the first step in dumping, reinitializing, and reloading the server database if a catastrophic error occurs (recovery log corruption, for example), and the DSMSERV RESTORE DB command cannot be used. No other server activity is allowed during DSMSERV DUMPDB command processing.

After DSMSERV DUMPDB processing is complete, perform the following steps:

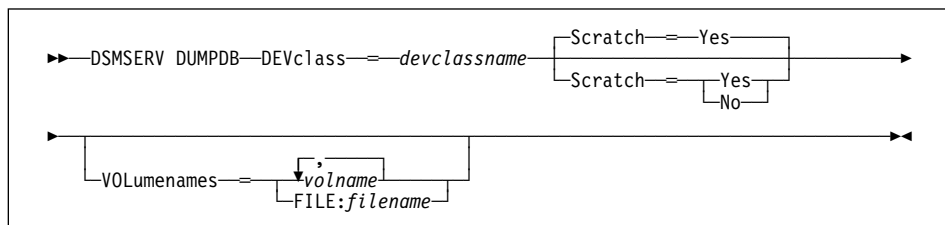
1. Issue the DSMSERV INSTALL command to reinitialize the database and recovery log.
2. Issue the DSMSERV LOADDDB command to reload the database.
3. If necessary, issue the DSMSERV AUDITDB command to locate and correct any database inconsistencies.

**Note:** A message at the end of the output from the DSMSERV LOADDDB command indicates whether a DSMSERV AUDITDB command is required. If the server was quiesced when it was halted, the DSMSERV AUDITDB command is not required.

The DSMSERV DUMPDB command requires a device configuration file. The server options file that is created during installation includes a DEVCONFIG option that specifies a default file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. The device configuration file contains a copy of the device class, library, and drive definitions that exist in your database.

Because the DSMSERV DUMPDB command requires a device configuration file, the file specified with the DEVCONFIG option must be available and must contain at least the device class specified for use with this command, along with any necessary library and drive definitions.

### Syntax



### Parameters

**DEVclass=devclassname**

Specifies the device class to which the database information is to be written. This parameter is required.

You must have a device configuration file that includes the definition for this device class, and any libraries and drives needed. The server options file that is created during installation includes a DEVCONFIG option that specifies a default

## DSMSERV DUMPDB

file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. If the device configuration file has been lost, do one of the following:

- If the server can be restarted, restart the server and issue the BACKUP DEVCONFIG command.
- If the server cannot be restarted, create the device configuration file manually with an editor. For information on how to create a device configuration file manually, see *ADSM Administrator's Guide*.

### **Scratch**=*scratchvalue*

Specifies that scratch volumes can be used for dumping the database. The default value is YES.

#### **Yes**

Specifies that scratch volumes can be used for dumping the database.

**Note:** You cannot specify SCRATCH=YES if the device class you specified with this command is associated with a SCSI library.

To determine the library type of the library, issue the the QUERY LIBRARY command and specify FORMAT=DETAILED.

If you include a list of volumes on the VOLUMENAMES parameter, then scratch volumes are used only if there is not enough space to dump the database on the volumes specified. If the device type associated with the specified device class is FILE, file names for the scratch volumes are generated based on a time stamp.

#### **No**

Specifies that scratch volumes cannot be used for dumping the database. You must include a list of volumes on the VOLUMENAMES parameter to contain all of the database data.

### **VOLumenames**=*volname*

Specifies one or more volumes to use to dump the database. This parameter is optional, but must be specified if SCRATCH=NO. If you do not specify this parameter and SCRATCH=YES is specified or assumed, scratch volumes are used.

ADSM does not record the use of volumes by the DSMSERV DUMPDB command in the volume history file. Therefore, you must record the volume names used and specify them in the exact same order on a future DSMSERV LOADDB command.

Possible values are:

#### *volname*

Specifies the names of one or more volumes to use to dump the database. You can specify more than one volume by separating each volume name with a comma, with no intervening spaces. You must list the volumes in the order in which you want the volumes to be used.

## DSMSERV DUMPDB

For example, if you want to use the volumes TAPE01, TAPE02, and TAPE03, specify the following:

```
VOLUMENAMES=TAPE01,TAPE02,TAPE03
```

**FILE:***filename*

Specifies the name of a file that contains a list of the volumes to use to dump the database. The file name can be any valid OS/2 file name. In the file, each volume name must be on a separate line. You must list the volumes in the order in which you want the volumes to be used.

For example, if you want to use the volumes TAPE01, TAPE02, and TAPE03, create a file named VOLLIST that contains these lines:

```
TAPE01  
TAPE02  
TAPE03
```

You can then specify the volumes for the dump command as follows:

```
VOLUMENAMES=FILE:VOLLIST
```

## Usage Notes

Consider the following factors before dumping the database:

- When dumping the database data to sequential access media, you need to estimate how much media is required to hold the data. If the server cannot initialize, a reasonable size estimate is equal to the size of your existing database volumes.

If you can bring the server up, you can estimate the size by following these steps:

- Step 1. Issue the QUERY DB command.
- Step 2. From the output of the QUERY DB command, multiply the number of used pages by the page size (4096 bytes) to determine the number of bytes currently occupied by the database.
- Step 3. Divide the product of step 2 by the capacity of a single medium to determine an estimate of the number of media required.

- Record the order in which the volumes are mounted during dump processing. The volume order is important during recovery processing so that the volumes can be remounted in the same order. To ensure that volumes are mounted in the correct order, prelabel the volumes with information that indicates the order in which they have been mounted. For example, label tapes as DSM001, DSM002, DSM003, and so on to indicate the order by which data is stored on the tape volumes.
- When dumping, you can use scratch volumes to ensure that there is sufficient space to store the database data. If you use scratch volumes, be careful to record the label names and sequence for each volume mounted during the dump process. The volumes used by the DSMSERV DUMPDB command are *not* recorded in the sequential volume history information.
- The server recovery log is not accessed during the dump process. Therefore, database entries that were not yet written to the database when the dump is

## DSMSERV DUMPDB

performed are not recorded. During recovery from a catastrophic failure, the most recent database updates may not be recoverable.

### Examples

#### Task

Dump the existing ADSM database to an 8mm tape named DB0001.

For this example the following steps are required:

- Step 1. Halt the server.
- Step 2. Ensure that the DEVCONFIG option has been specified in the server options file. The device configuration file specified with this option must exist and must contain the device class, library, and drive definitions needed for the dump operation.
- Step 3. Prelabel the tape using the dsmlabel utility. For example:  
`dsmlabel -drive=tape05$`
- Step 4. Issue the DSMSERV DUMPDB command.

**Command:** `dsmserv dumpdb devclass=8mmc class volnames=DB0001`

## DSMSERV DUMPDB

### Output:

```
ANR7800I DSMSERV generated at 11:27:09 on Jun 17 1996.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
ANR4000I DUMPDB: Database dump process started.
ANR8326I 001: Mount 8MM volume DB0001 R/W in drive 8MMDRIVE
(TAPE05$) of library 8MMLIB within 60 minutes.
ANR8335I 001: Verifying label of 8MM volume DB0001 in drive 8MMDRIVE
(TAPE05$).
ANR8328I 001: 8MM volume DB0001 mounted in drive 8MMDRIVE (TAPE05$).
ANR1360I Output volume DB0001 opened (sequence number 1).
ANR1361I Output volume DB0001 closed.
ANR8336I Verifying label of 8MM volume DB0001 in drive 8MMDRIVE
(TAPE05$).
ANR1360I Output volume DB0001 opened (sequence number 1).
ANR1361I Output volume DB0001 closed.
ANR8336I Verifying label of 8MM volume DB0001 in drive 8MMDRIVE
(TAPE05$).
ANR8468I 8MM volume DB0001 dismounted from drive 8MMDRIVE (TAPE05$)
in library 8MMLIB.
ANR4031I DUMPDB: Copied 31 database pages.
ANR4033I DUMPDB: Copied 6 bit vectors.
ANR4034I DUMPDB: Encountered 0 bad database pages.
ANR4036I DUMPDB: Copied 193 database entries.
ANR4037I DUMPDB: 36 Kilobytes copied.
ANR4001I DUMPDB: Database dump process completed.
```

## DSMSERV INSTALL

---

### DSMSERV INSTALL (Install the Database and Log)

Use the DSMSERV INSTALL command to run the server's install mode to initialize the server's database and recovery log. No other server activity is allowed during DSMSERV INSTALL command processing.

For improved system performance, use the high performance file system (HPFS) instead of the file allocation table (FAT) file system when executing this command.

#### Syntax

```
▶▶ dsmserv install -nlog -lname -lsize -ndb -dbname -dbsize ▶▶
```

#### Parameters

<i>nlog</i>	specifies the number of files to be used by the recovery log.
<i>lname</i>	specifies the name of the file being used by the recovery log. This file must <i>not</i> already exist.
<i>lsize</i>	specifies the size of the recovery log file in megabytes. The minimum size is 9MB.
<i>ndb</i>	specifies the number of files to be used by the database.
<i>dbname</i>	specifies the name of the file being used by the database. This file must <i>not</i> exist prior to using the DSMSERV INSTALL command.
<i>dbsize</i>	specifies the database size in megabytes. The minimum size is 5MB.

#### Usage Notes

- *lsize* and *dbsize* are positive integers, in MB. Allocated sizes will be in multiples of 4MB, plus 1MB of overhead.
- If you use an asterisk. (\*) for the file size, the size will equal the remaining space minus 32K. The minimum requirements for size, which are 9MB for logs and 5MB for databases, must still be met.
- The install command will fail if there is not enough space available.

## DSMSERV INSTALL

### Examples

An example of issuing the DSMSERV INSTALL command to allocate 9 megabytes each for two recovery log files and 5 megabytes for one database follows.

**Command:** dsmserv install 2 log1 9 log2 9 1 dbvol1 5

**Output:**

```
ANR7800I DSMSERV generated at 11:27:09 on Jun 17 1996.  
Allocated space for D:\adsmv2\log1: 9437184 bytes  
Allocated space for D:\adsmv2\log2: 9437184 bytes  
Allocated space for D:\adsmv2\dbvol1: 5242880 bytes  
ANR0300I Recovery log format started; assigned capacity 16 megabytes.  
ANR0301I Recovery log format in progress; 4 megabytes of 16.  
ANR0301I Recovery log format in progress; 8 megabytes of 16.  
ANR0301I Recovery log format in progress; 12 megabytes of 16.  
ANR0301I Recovery log format in progress; 16 megabytes of 16.  
ANR0302I Recovery log formatting took 25498 milliseconds.  
ANR0303I Format rate: 160.6 pages/second.  
ANR0304I Page service time: 6.2 ms.  
ANR0305I Recovery log format complete.  
ANR0306I Recovery log volume mount in progress.  
ANR0353I Recovery log analysis pass in progress.  
ANR0354I Recovery log redo pass in progress.  
ANR0355I Recovery log undo pass in progress.  
ANR0352I Transaction recovery complete.  
ANR0992I ADSM server installation complete.
```



## DSMSERV LOADDDB

### DSMSERV LOADDDB (Reload the Database)

Use the DSMSERV LOADDDB command to reload a dumped ADSM database to a newly installed ADSM server. The database must have been dumped by DSMSERV DUMPDB command. You must reinstall ADSM using the DSMSERV INSTALL command prior to issuing the DSMSERV LOADDDB command. No other server activity is allowed during DSMSERV LOADDDB command processing.

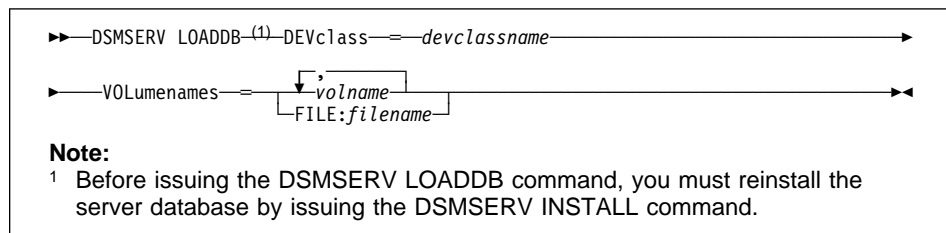
When recovering from a failure, it is recommended that you reinstall your server using a new database and log so that the original database and log volumes remain intact in case the dump and load process needs to be repeated.

A message at the end of the output from the DSMSERV LOADDDB indicates whether a DSMSERV AUDITDB command is required. If the server was quiesced when it was halted before the DSMSERV DUMPDB command was issued, the DSMSERV AUDITDB command is not required. Otherwise, you must issue the DSMSERV AUDITDB command to ensure that the database is returned to a synchronized state after it is reloaded.

The DSMSERV LOADDDB command requires a device configuration file. The server options file that is created during installation includes a DEVCONFIG option that specifies a default file named devcnfg.out. You can specify other device configuration files with additional DEVCONFIG options. The device configuration file contains a copy of the device class, library, and drive definitions that exist in your database.

Because the DSMSERV LOADDDB command requires a device configuration file, the file specified with the DEVCONFIG option must be available and must contain at least the device class specified for use with this command, along with any necessary library and drive definitions.

### Syntax



### Parameters

#### DEVclass=*devclassname*

Specifies the device class from which the database information is to be read. This parameter is required.

You must have a device configuration file that includes the definition for this device class, and any libraries and drives needed. The server options file that is created during installation includes a DEVCONFIG option that specifies a default file named devcnfg.out. You can specify other device configuration files with

## DSMSERV LOADDDB

additional DEVCONFIG options. If the device configuration file has been lost , do one of the following:

- If the server can be restarted, restart the server and issue the BACKUP DEVCONFIG command.
- If the server cannot be restarted, create the device configuration file manually with an editor. For information on how to create a device configuration file manually, see *ADSM Administrator's Guide*.

### **VOL**umenames=*volname*

Specifies one or more volumes from which to load the database. This parameter is required. You must specify the same list of volumes in the same order as that used for the DSMSERV DUMPDB command.

Possible values are:

#### *volname*

Specifies the names of one or more volumes from which to load the database. You can specify more than one volume by separating each volume name with a comma, with no intervening spaces. You must list the volumes in the order in which you want the volumes to be loaded (that is, the same order as that used for the DSMSERV DUMPDB command).

For example, if you want to use the volumes TAPE01, TAPE02, and TAPE03, specify the following:

```
VOLUMENAMES=TAPE01,TAPE02,TAPE03
```

### **FILE:***filename*

Specifies the name of a file that contains a list of the volumes from which to load the database. The file name can be any valid OS/2 file name. In the file, each volume name must be on a separate line. You must list the volumes in the order in which you want the volumes to be loaded (that is, the same order as that used for the DSMSERV DUMPDB command).

For example, if you want to use the volumes TAPE01, TAPE02, and TAPE03, create a file named VOLLIST that contains these lines:

```
TAPE01  
TAPE02  
TAPE03
```

You can then specify the volumes for the load command as follows:

```
VOLUMENAMES=FILE:VOLLIST
```

## Usage Notes

- Before you issue the DSMSERV LOADDDB command, save your current sequential volume history to a file for future reference. The DSMSERV LOADDDB process causes the sequential volume history information to be reset.
- If you must issue the DSMSERV AUDITDB command, ensure that it has completed processing before you start the server.

## DSMSERV LOADDDB

- If, since the dump of the database you are reloading, there have been any activities that involve the storage pool volumes, audit the volumes by using the AUDIT VOLUME command. You should issue the AUDIT VOLUME command for all of your disk volumes. The volume history information that you saved earlier can help you identify which sequential volumes need to be audited. You should audit all volumes that have been reused since the time of the original dump of the database. Any volumes that were deleted from ADSM are also recorded in the sequential volume history information. If these volumes can still be obtained, you can try auditing these volumes. If these volumes cannot be obtained, you should delete references to these volumes in the database using the DELETE VOLUME command with the DISCARDDATA=YES option.

### Examples

#### Task

Load the ADSM database from the previously dumped DB0001 tape.

For this example the following steps are required:

- Step 1. Ensure that the DEVCONFIG option has been specified in the server options file. The device configuration file specified with this option must exist and must contain the device class, library, and drive definitions needed for the load operation.
- Step 2. Issue the DSMSERV LOADDDB command.

**Command:** `dsmserv loaddb devclass=8mmclass volnames=DB0001`

## DSMSERV LOADDDB

### Output:

```
ANR7800I DSMSERV generated at 11:27:09 on Jun 17 1996.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 16 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR4003I LOADDDB: Database load process started.
ANR8326I 001: Mount 8MM volume DB0001 R/O in drive 8MMDRIVE
(TAPE05$) of library 8MMLIB within 60 minutes.
ANR8335I 001: Verifying label of 8MM volume DB0001 in drive 8MMDRIVE
(TAPE05$).
ANR8328I 001: 8MM volume DB0001 mounted in drive 8MMDRIVE (TAPE05$).
ANR1363I Input volume DB0001 opened (sequence number 1).
ANR4038I LOADDDB: Loading database information dumped on 06/18/1996
at 11:25:13.
ANR1364I Input volume DB0001 closed.
ANR8468I 8MM volume DB0001 dismounted from drive 8MMDRIVE (TAPE05$)
in library 8MMLIB.
ANR4032I LOADDDB: Copied 31 database records.
ANR4033I LOADDDB: Copied 6 bit vectors.
ANR4035I LOADDDB: Encountered 0 bad database records.
ANR4036I LOADDDB: Copied 193 database entries.
ANR4037I LOADDDB: 36 Kilobytes copied.
ANR4004I LOADDDB: Database load process completed.
ANR4405I LOADDDB: Loaded an inconsistent dump image - a database audit
(AUDITDB) IS REQUIRED.
```

## DSMSERV AUDITDB

### DSMSERV AUDITDB (Verify Database Information)

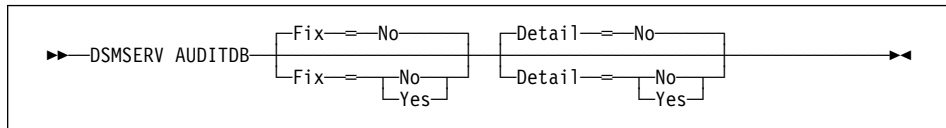
Use the DSMSERV AUDITDB command when required to ensure that the database is returned to a synchronized state after it is reloaded via the DSMSERV LOADDDB command. A message at the end of the output from the DSMSERV LOADDDB command indicates whether this command is required.

You must issue this command prior to starting the server. To successfully use this command, one of the following conditions must be true:

- The recovery log mode was set to normal before the server was halted (by using the SET LOGMODE command)
- A backup of the database was completed just before the server was halted

No other server activity is allowed during DSMSERV AUDITDB command processing.

### Syntax



### Parameters

#### **Fix**=*fixvalue*

Specifies how ADSM resolves any inconsistencies in the database. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that ADSM reports, but does not fix, any logical inconsistencies found.

If the audit finds inconsistencies when the command is run with FIX=NO specified, before making the server available for production work you must re-issue the command specifying FIX=YES. Because AUDITDB must be run with FIX=YES to recover the database, the recommended usage in a recovery situation is FIX=YES the first time.

#### **Yes**

Specifies that ADSM fixes any inconsistencies and issues messages indicating the corrective actions.

#### **Detail**=*detailvalue*

Specifies the level of database testing to be performed by ADSM. This parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that ADSM only tests referential integrity of the database.

## DSMSERV AUDITDB

### Yes

Specifies that ADSM tests referential integrity of the database and the integrity of each individual database entry.

## Usage Notes

If, prior to ADSM server reinstallation, STANDARD policy objects were deleted, during DSMSERV AUDITDB command processing messages are issued indicating that an instance count does not agree with actual data. You can delete the predefined STANDARD policy objects; however, all STANDARD policy objects are restored during ADSM server reinstallation. DSMSERV AUDITDB does not delete the STANDARD policy objects; however, it does synchronize the instance count to include the restored STANDARD policy objects.

## Examples

### Task 1

Audit the ADSM database for inconsistencies, but do not fix inconsistencies if they exist.

**Command:** dsmserv auditdb fix=no

### Output:

```
ANR0900I Processing options file dsmserv.opt.

ANR7800I DSMSERV generated at 11:27:09 on Jun 17 1996.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 16 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR4140I AUDITDB: Database audit process started.
ANR4075I AUDITDB: Auditing policy definitions.
ANR4040I AUDITDB: Auditing client node and administrator definitions.
ANR4135I AUDITDB: Auditing central scheduler definitions.
ANR2833I AUDITDB: Auditing license definitions.
ANR4136I AUDITDB: Auditing server inventory.
ANR4137I AUDITDB: Auditing inventory file spaces.
ANR4307I AUDITDB: Auditing inventory external space-managed objects.
ANR4138I AUDITDB: Auditing inventory backup objects.
ANR4139I AUDITDB: Auditing inventory archive objects.
ANR4310I AUDITDB: Auditing inventory space-managed objects.
ANR4230I AUDITDB: Auditing data storage definitions.
ANR4264I AUDITDB: Auditing file information.
ANR4265I AUDITDB: Auditing disk file information.
ANR4266I AUDITDB: Auditing sequential file information.
ANR4256I AUDITDB: Auditing data storage definitions for disk volumes.
ANR4263I AUDITDB: Auditing data storage definitions for sequential
volumes.
ANR4210I AUDITDB: Auditing physical volume repository definitions.
ANR4141I AUDITDB: Database audit process completed.
```

## DSMSERV AUDITDB

### Task 2

Audit the ADSM database for inconsistencies and fix any inconsistencies if they exist.

**Command:** dsmserv auditdb fix=yes

#### Output:

```
ANR0900I Processing options file dsmserv.opt.

ANR7800I DSMSERV generated at 16:44:34 on June 21 1996.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 16 megabytes.
ANR0201I Database assigned capacity is 12 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR4140I AUDITDB: Database audit process started.
ANR4075I AUDITDB: Auditing policy definitions.
ANR4087I AUDITDB: Invalid node 88 found assigned to domain STANDARD
reference will be deleted.
ANR4077I AUDITDB: Invalid client node count detected for policy domain
STANDARD count will be corrected.
ANR4096E AUDITDB: An invalid management class identifier (22) was
encountered for management class STANDARD in policy set STANDARD,
domain STANDARD (expected 15) - the correct identifier will be stored.
ANR4090E AUDITDB: Active Policy set for domain STANDARD does not contain
a valid default management class - policy will fail for this domain.
Please correct the ACTIVE policy set.
ANR4040I AUDITDB: Auditing client node and administrator definitions.
ANR4042I AUDITDB: Node DAVE is assigned to Domain STANDARD, but not
verified by the Policy Manager - attempting to assign the node to
domain STANDARD.
ANR4066I AUDITDB: Administrator 99 not found in the server database
- reference will be removed.
ANR4135I AUDITDB: Auditing central scheduler definitions.
ANR2833I AUDITDB: Auditing license definitions.
ANR4136I AUDITDB: Auditing server inventory.
ANR4137I AUDITDB: Auditing inventory file spaces.
ANR4307I AUDITDB: Auditing inventory external space-managed objects.
ANR4138I AUDITDB: Auditing inventory backup objects.
ANR4139I AUDITDB: Auditing inventory archive objects.
ANR4310I AUDITDB: Auditing inventory space-managed objects.
ANR4230I AUDITDB: Auditing data storage definitions.
ANR4264I AUDITDB: Auditing file information.
ANR4265I AUDITDB: Auditing disk file information.
ANR4266I AUDITDB: Auditing sequential file information.
ANR4256I AUDITDB: Auditing data storage definitions for disk volumes.
ANR4306I AUDITDB: 25000 Records processed.
ANR4263I AUDITDB: Auditing data storage definitions for sequential
volumes.
ANR4210I AUDITDB: Auditing physical volume repository definitions.
ANR4141I AUDITDB: Database audit process completed.
```

### Task 3

The following example is of output where no inconsistencies were found when fix=no was specified.

**Command:** dsmserv auditdb fix=no

## DSMSERV AUDITDB

### Output:

```
ANR7800I DSMSERV generated at 13:38:18 on June 21 1996.
ANR7801I Subsystem process ID is 999.
ANR0900I Processing options file dsmserv.opt.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 52 megabytes.
ANR0201I Database assigned capacity is 328 megabytes.
ANR0306I Recovery log volume mount in progress.
ANR0353I Recovery log analysis pass in progress.
ANR0354I Recovery log redo pass in progress.
ANR0355I Recovery log undo pass in progress.
ANR0352I Transaction recovery complete.
ANR4140I AUDITDB: Database audit process started.
ANR4075I AUDITDB: Auditing policy definitions.
ANR4040I AUDITDB: Auditing client node and administrator definitions.
ANR4135I AUDITDB: Auditing central scheduler definitions.
ANR2833I AUDITDB: Auditing license definitions.
ANR4136I AUDITDB: Auditing server inventory.
ANR4137I AUDITDB: Auditing inventory file spaces.
ANR4307I AUDITDB: Auditing inventory external space-managed objects.
ANR4138I AUDITDB: Auditing inventory backup objects.
ANR4139I AUDITDB: Auditing inventory archive objects.
ANR4310I AUDITDB: Auditing inventory space-managed objects.
ANR4230I AUDITDB: Auditing data storage definitions.
ANR4264I AUDITDB: Auditing file information.
ANR4265I AUDITDB: Auditing disk file information.
ANR4266I AUDITDB: Auditing sequential file information.
ANR4256I AUDITDB: Auditing data storage definitions for disk volumes.
ANR4263I AUDITDB: Auditing data storage definitions for sequential
volumes.
ANR4210I AUDITDB: Auditing physical volume repository definitions.
ANR4141I AUDITDB: Database audit process completed.
```



---

## Appendix E. Tailoring Server Options

ADSM provides a server options file that allows you to customize the following options:

- Communications
- Client-server operations
- Date, number, time, and language
- Database and recovery log
- Message Options
- Requesting information about server options

---

### Modifying the Server Options File

At installation, ADSM provides a sample server options file with minimal options to start the server named `dsmserv.opt`. The `dsmserv.smp` file (also provided at installation) contains the format of the options file as well as all of the default settings. You can change all the options in the `dsmserv.smp` file. If you do so, you will need to rename this file to `dsmserv.opt`.

To activate an option within the server options file, you must remove the `*>>>` that appears prior to the option. Any options preceded by an `*>>>` will be ignored by the server.

At server initialization, the server reads the server options file. If you make any changes to the server options file, you must stop and start the server to have the server read the updated server options file.

### Setting Communication Options

Communication protocols are available to communicate with clients. Table 180 displays the communication methods supported for OS/2 servers.

---

*Table 180 (Page 1 of 2). Supported Communication Methods for an ADSM Server on OS/2*

Communication Method	Options Specified
APPC(SNALU6.2)	<ul style="list-style-type: none"><li>• LUsername, see "LUsername" on page 687</li><li>• SNABuffersize, see "SNABuffersize" on page 694</li></ul>
IPXSPX	<ul style="list-style-type: none"><li>• IPXBuffersize, see "IPXBuffersize" on page 685</li><li>• IPXSocket, see "IPXSocket" on page 685</li></ul>
NAMED PIPES	<ul style="list-style-type: none"><li>• NAMEdpipename, see "NAMEdpipename" on page 691</li></ul>

Table 180 (Page 2 of 2). Supported Communication Methods for an ADSM Server on OS/2

Communication Method	Options Specified
NETBIOS	<ul style="list-style-type: none"> <li>• LANAdapter, see “LANAdapter” on page 686</li> <li>• NETBIOSBuffersize, see “NETBIOSBuffersize” on page 692</li> <li>• NETBiosname, see “NETBiosname” on page 692</li> <li>• NETBIOSSession, see “NETBIOSSession” on page 693</li> </ul>
SHAREDMEMORY	<ul style="list-style-type: none"> <li>• SHMQUEUEENAME, see “SHMQUEUEENAME” on page 694</li> </ul>
TCPIP	<ul style="list-style-type: none"> <li>• TCPPort, see “TCPPort” on page 696</li> <li>• TCPNodelay, see “TCPNodelay” on page 695</li> <li>• TCPWindowSize, see “TCPWindowSize” on page 696.</li> </ul>
NONE	<ul style="list-style-type: none"> <li>• This communication method does not allow users to connect to the server. This method is specified as COMMETHOD NONE.</li> </ul>

## Setting Client-Server Options

Use the options identified in Table 181 to set the length of time the server waits for communication with an active or idle client, limit the number of simultaneous client sessions with each server, and set the interval between file inventory expiration runs.

Table 181 (Page 1 of 2). Client-Server Options

Command	Description	Reference
COMMOPENTIMEOUT	Specifies the number of seconds the server waits for a response from a client when the server is initiating the communication. (Server-prompted scheduling)	“COMMOpentimeout” on page 681
COMMTIMEOUT	Specifies the number of seconds the server waits for a response from a client before timing out the client session.	“COMMTIMEout” on page 682
EXPINTERVAL	Specifies the interval in hours between automatic inventory expiration runs by the ADSM server.	“EXPINTERval” on page 683
EXPQUIET	Specifies whether ADSM sends detailed informational messages.	“EXPQUIet” on page 684
IDLETIMEOUT	Specifies the number of minutes the server allows a client session to remain idle before timing out the client session.	“IDLETIMEout” on page 685

Table 181 (Page 2 of 2). Client-Server Options

Command	Description	Reference
MAXSESSIONS	Specifies the maximum number of simultaneous client sessions with the server.	"MAXSessions" on page 687
NOAUDITSTORAGE	Specifies that as part of the license audit operation, the server calculates, by node, the amount of backup, archive, and space management storage in use.	"NOAUDITStorage" on page 693
TXNGROUPMAX	Specifies the number of files that are transferred as a group between a client and the server between transaction commit points.	"TXNGroupmax" on page 697

## Setting Date, Number, Time, and Language Options

Use the options identified in Table 182 to specify display formats for the dates, times, numbers, and national language.

Table 182. Date, Number, Time, and Language Options

Command	Description	Reference
DATEFORMAT	Specifies the format by which dates are displayed.	"DATEformat" on page 682
LANGUAGE	Specifies which national language is used to present client messages.	"LANGUage" on page 686
NUMBERFORMAT	Specifies the format by which numbers are displayed.	"NUMberformat" on page 693
TIMEFORMAT	Specifies the format in which times are displayed.	"TIMEformat" on page 696

## Setting Database and Recovery Log Options

Use the options in Table 183 for:

- Tailoring the size of the database and recovery log buffer
- Recovery of the server
- Specifying when recovery log checkpoints should occur
- Specifying how mirrored volumes are to be accessed

Table 183 (Page 1 of 2). Database and Recovery Log Options

Command	Description	Reference
BUFPOOLSIZE	Specifies the size of database buffer pool in kilobytes (KB).	"BUFPoolsize" on page 681
DEVCONFIG	Specifies a file in which you want ADSM to store backup copies of device configuration information.	"DEVCONFig" on page 683

Table 183 (Page 2 of 2). Database and Recovery Log Options

Command	Description	Reference
LOGPOOLSIZE	Specifies the size of recovery log buffer pool in kilobytes (KB).	"LOGPoolsize" on page 687
MIRRORREAD	Specifies how mirrored volumes are to be accessed when the server reads a recovery log or database page during normal processing.	"MIRRORRead" on page 689
MIRRORWRITE	Specifies how mirrored volumes are to be accessed when the server writes pages to the recovery log or database during normal processing.	"MIRRORWrite" on page 690
MOVEBATCHSIZE	Specifies the number of files that are to be moved and grouped in a batch, within the same server transaction.	"MOVEBatchsize" on page 690
MOVESIZETHRESH	Specifies, in megabytes, a threshold for the amount of data moved as a batch, within the same server transaction.	"MOVESizethresh" on page 691
STATUSMSGCNT	Specifies the number of records (times 1000) that will be processed between messages during DSMSERV DUMPDB and DSMSERV LOADDB commands.	"STATUSmsgcnt" on page 695
VOLUMEHISTORY	Specifies files to be automatically updated whenever server sequential volume history information is changed.	"VOLUMEHISTORY" on page 698

## Setting Message Options

Message options in Table 184 provide you with more flexibility in the way the ADSM server issues messages to the operator console. You can highlight messages, suppress messages, or specify a different routing code for messages. The options do not affect messages issued before the options file is read, or the terminating message is issued by ADSM.

Table 184. Message Options

Option	Description	Reference
MSGSUPPRESS	Specifies the message level to be suppressed.	"MSGSuppress" on page 688.
MESSAGEFORMAT	Specifies whether or not a message number is displayed in all lines of a multi-line message.	"MESSAGEformat" on page 688

---

## Requesting Information about Server Options

You can display the option settings specified in the server options file.

To query for server options, enter:

```
Query OPTion
```

For more information, see the QUERY OPTIONS command.

---

## BUFPoolsize

Specifies the size of the database buffer pool in kilobytes (KB). A large buffer pool means that database pages remain longer in memory cache, and ADSM requires fewer input/output operations to data storage. However, a large buffer pool also requires more memory.

### Syntax

```
▶▶—BUFPoolsize—kilobytes————▶▶
```

### Parameters

*kilobytes*

Specifies the size of the database buffer pool. The default value is 512KB. The minimum value is 256KB. The maximum value is limited only by available virtual memory size.

### Examples

```
BUFPoolsize 512
BUFP        256
```

---

## COMMOpentimeout

Specifies the maximum number of seconds the server waits for a response from a client when trying to initiate a conversation.

If this maximum is reached during server prompted scheduling, the server continues with the next client.

### Syntax

```
▶▶—COMMOpentimeout—seconds————▶▶
```

### Parameters

*seconds*

Specifies the maximum number of seconds that a server waits for a client response during server initiated communications. The default value is 20 seconds. The minimum value is 1 second.

## Examples

```
COMMOpentimeout 20
COMMO            60
```

---

## COMMTimeout

Specifies the maximum number of seconds the server waits, during a database update transaction, for an expected message from a client, before terminating the session with the client.

The server terminates the session to release communication resources as soon as possible, and to ensure that database locks are not held for undue periods of time.

## Syntax

```
►►—COMMTimeout—seconds—◄◄
```

## Parameters

*seconds*

Specifies the maximum number of seconds that a server waits for a client response. The default value is 60 seconds. The minimum value is 1 second.

## Examples

```
COMMTimeout 60
COMMT      120
```

---

## DATEformat

Specifies the format by which dates are displayed by the server.

## Syntax

```
►►—DATEformat—number—◄◄
```

## Parameters

*number*

Select a number from 1 to 5 to identify the date format used by the server. The default value is 1.

```
1          MM/DD/YYYY
```

2	DD-MM-YYYY
3	YYYY-MM-DD
4	DD.MM.YYYY
5	YYYY.MM.DD

## Examples

DATEformat 4

---

## DEVCONFig

Specifies the name of a file in which you want ADSM to store a backup copy of device configuration information.

ADSM stores the following information in the device configuration file:

- Device class definitions created by using the DEFINE DEVCLASS command
- Drive definitions created by using the DEFINE DRIVE command
- Drive definitions created by using the DEFINE LIBRARY command

A server options file that is created during installation includes a DEVCONFIG option that specifies a default device configuration file (devcnfg.out). You can include additional DEVCONFIG options in the server options file. When you use multiple DEVCONFIG options, ADSM automatically updates and stores a backup copy of device configuration information in each file you specify.

## Syntax

▶▶—DEVCONFig—*file\_name*————▶▶

## Parameters

*file\_name*

Specifies the name of a file in which you want ADSM to store a backup copy of device configuration information.

## Examples

DEVCONFig devices.sav

---

## EXPINterval

Specifies the interval in hours between automatic inventory expiration runs by the ADSM server. Inventory expiration removes client backup and archive file copies from the server as specified by the management classes to which the client files are bound. By default, the ADSM server automatically runs inventory expiration each hour (the interval is one hour between automatic expiration runs).

You can also use the EXPIRE INVENTORY command to start inventory expiration. Expiration can make space available in your storage pools for additional client backup or archive files. If expiration is not executed periodically, storage pool space is not reclaimed from expired client files, and the ADSM server requires more storage space than required by policy.

## Syntax

```
►►—EXPINterval—hours—————►◄
```

## Parameters

*hours*

Specifies the number of hours between automatic inventory expiration runs. The minimum value permitted is 0, where automatic expiration will not execute and must be started with the EXPIRE INVENTORY command. The maximum value permitted is 336 hours (14 days).

## Examples

```
EXPINterval 5  
EXPIN      0
```

---

## EXPQUIet

Specifies whether ADSM sends detailed informational messages when you have made policy changes such as deleting a copy group or management class that cause files to be examined for expiration based on a copy group in the default management class for the policy set or the retention grace period for the domain.

## Syntax

```
►►—EXPQUIet—  NO  YES —————►◄
```

## Parameters

**NO**

Specifies that ADSM sends only minimal summary messages and only for files that have expired.

**YES**

Specifies that ADSM sends detailed messages.

## Examples

```
EXPQUIet No
```



---

## IDLETimeout

Specifies the amount of time, in minutes, that a server waits for a client to initiate communication. If the client does not initiate communication with the server within the timeframe specified, the client session is terminated.

### Syntax

```
▶▶ IDLETimeout minutes ◀◀
```

### Parameters

*minutes*

Specifies the maximum number of minutes that a server waits for an idle client. The default value is 15 minutes. The minimum value is 1 minute.

### Examples

```
IDLETimeout 15
IDLET      30
```

---

## IPXBufferSize

Specifies the IPX send buffer size.

### Syntax

```
▶▶ IPXBufferSize value ◀◀
```

### Parameters

*value*

Specifies the size (in kilobytes) of the IPX/SPX communications buffer. You can specify a value from 1 kilobyte to 32 kilobytes. The default value is 32 kilobytes.

### Examples

```
IPXBufferSize 32
```

---

## IPXSocket

Specifies the IPX socket of an ADSM server.

### Syntax

```
▶▶ IPXSocket value ◀◀
```

## Parameters

*value*

Specifies the IPX socket number (hex value). The default is 8522. You can specify a value from 0 to ffff.

## Examples

```
IPXSocket 8522
```

---

## LANAdapter

Specifies a network adapter number on which the server communicates when the NetBIOS communication method is specified.

## Syntax

```
▶▶—LANAdapter—value————▶▶
```

## Parameters

*value*

Specifies the number of the LAN adapter to use with the ADSM server for NetBIOS. The minimum value of the LAN adapter is 0 and the maximum value is 3. Up to 4 LANAdapter statements may be specified. The default is 0.

## Examples

```
LANAdapter 0
```

---

## LANGuage

Specifies the national language used to present client messages. Currently, only American English is supported.

## Syntax

```
▶▶—LANGuage—language————▶▶
```

## Parameters

*language*

Specifies the language to use for client messages. AMENG is the default.

## Examples

```
LANG AMENG
```

---

## LOGPoolsize

Specifies the size of the recovery log buffer pool size in kilobytes. A large buffer pool may increase the rate by which recovery log transactions are committed to the database, but it also requires more memory.

### Syntax

```
▶▶—LOGPoolsize—kilobytes—▶▶
```

### Parameters

*kilobytes*

Specifies the size of the recovery log buffer pool. The default value is 128KB. The minimum value is 128KB. The maximum value is limited only by available virtual memory size.

### Examples

```
LOGPoolsize      128
```

---

## LUname

Specifies the name ADSM uses for APPC communications on the network. A maximum of four LU names can be specified. The maximum length of an LU name is eight characters.

### Syntax

```
▶▶—LUname—value—▶▶
```

### Parameters

*value*

Specifies the name ADSM uses for APPC communications on the network. A maximum of four LU names can be specified. The maximum length of an LU name is eight characters.

### Examples

```
LUname           DSMSEV1
```

---

## MAXSessions

Specifies the maximum number of simultaneous client sessions that can connect with the server.

## Syntax

```
▶▶—MAXSessions—numsessions—◀◀
```

## Parameters

*numsessions*

Specifies the maximum number of simultaneous client sessions. The default value is 25 client sessions. The minimum value is 2 client sessions. The maximum value is limited only by available virtual memory size or communication resources.

## Examples

```
MAXSessions 25
```

---

## MSGSuppress

Specifies the message level for suppression. Message suppression is done based on the severity of the message - ANRxxxxs, where s is severity and can be I,W,E,D and S. Suppression does not affect the response to console commands or the contents of the activity log. Message suppression is limited to Warning and below.

## Syntax

```
▶▶—MSGSuppress—number—◀◀
```

## Parameters

*number*

Select a number to identify the highest message level to be suppressed. The default value is 0.

- |          |                                      |
|----------|--------------------------------------|
| <b>0</b> | No messages are suppressed.          |
| <b>1</b> | All I messages are suppressed.       |
| <b>2</b> | All I and W messages are suppressed. |

## Examples

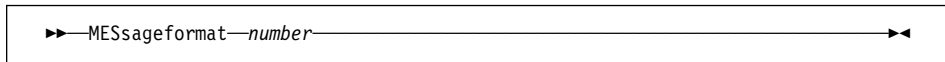
```
MSGSuppress 1
```

---

## MESSageformat

Specifies whether or not a message number is displayed in all lines of a multi-line message.

## Syntax



## Parameters

### *number*

Select a number to specify if a message number is to be displayed only on the first line of a multi-line message or is to be displayed on all lines. The default value is 1.

- 1** The message number for a message is displayed only at the first line of the message.
- 2** The message number for a message is displayed at all lines of a message.

## Examples

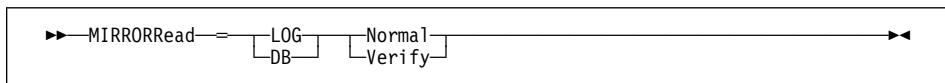
MESSageformat 2

---

## MIRRORRead

Specifies how mirrored volumes are to be accessed when the server reads a recovery log or database page during normal processing.

## Syntax



## Parameters

### **LOG**

Specifies that this is the mode to be used for reading recovery log pages

### **DB**

Specifies that this is the mode to be used for reading database pages

### **Normal**

Specifies only one mirrored volume is read to obtain the desired page. This is the default.

### **Verify**

Specifies that the server read all mirror volumes for a page every time a recovery log or database page is read. If an invalid page is encountered on a mirror volume, it is automatically re-synchronized with valid contents from another mirror volume. This mode decreases server performance since each mirror volume for the page is accessed on every read.

## Examples

```
MIRRORREAD LOG NORMAL
MIRRORREAD DB NORMAL
```

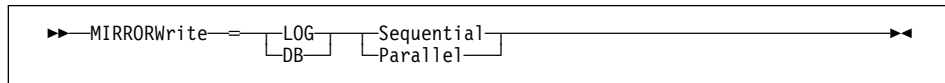
These are the default settings.

---

## MIRRORWrite

Specifies how mirrored volumes are accessed when the server writes pages to the recovery log or database during normal processing.

## Syntax



## Parameters

### LOG

Specifies that this is the mode to be used for writing recovery log pages

### DB

Specifies that this is the mode to be used for writing database pages

### Sequential

Specifies that one mirror is written to successfully before other mirrored volumes are directed to write their page. A decrease in server performance is experienced because a successful I/O must occur before the other mirrored writes can begin. This parameter is the default when *DB* is specified.

### Parallel

Specifies that all mirrors are written to in parallel. This mode allows all mirrors to obtain the new page at approximately the same time. This parameter is the default when *LOG* is specified. Note: If a system outage occurs at exactly the instant that each mirror is partially complete in writing its page, a partial write to each mirror could result.

## Examples

```
MIRRORWRITE LOG PARALLEL
MIRRORWRITE DB SEQUENTIAL
```

These are the default settings.

---

## MOVEBatchsize

Specifies the number of files that are to be moved and grouped together in a batch, within the same server transaction.

## Syntax

▶▶—MOVEBatchsize—*nnn*————▶▶

## Parameters

*nnn*

Specifies a number between 1 and 256. The default value is 40. This option works in conjunction with the MOVE Sizethresh option.

---

## MOVE Sizethresh

Specifies, in megabytes, a threshold for the amount of data moved as a batch, within the same server transaction. When this threshold is reached, no more files are added to the current batch, and a new transaction is started after the current batch is moved.

## Syntax

▶▶—MOVE Sizethresh—*nnn*————▶▶

## Parameters

*nnn*

Specifies a number between 1 and 500 (megabytes). The default value is 500 (megabytes). This option works in conjunction with the MOVE Batchsize option.

---

## NAMEd pipename

Specifies a communication method that allows processes to communicate with one another without having to know where the sender and receiver processes are located. The name acts like an alias, connecting the two processes regardless of whether they are on the same computer or across connected domains.

## Syntax

▶▶—NAMEd pipename—*value*————▶▶

## Parameters

*value*

Specifies the named pipes names for the server to use. Names pipes are ideal for running in an environment where client and server are on the same machine. No communication software is required and no setup is required.

## Examples

```
NAMEDpipeName \PIPE\DSMSERV
```

---

## NETBIOSBufferSize

Specifies the size of the NetBIOS send and receive buffers.

### Syntax

```
▶▶—NETBIOSBufferSize—value————▶▶
```

### Parameters

*value*

Specifies the size of NetBIOS send and receive buffers in kilobytes. The default buffer size is 32 KB. The minimum size is 1KB and the maximum size is 32KB.

## Examples

```
NETBIOSBufferSize 32
```

---

## NETBiosname

Specifies the NetBIOS session name of an ADSM server.

### Syntax

```
▶▶—NETBiosname—name————▶▶
```

### Parameters

*name*

Specifies the name to be used for communications on the network. This name must be unique across the network that includes the LAN requestors and other NetBIOS applications. This name can be a maximum of 16 characters and is case-sensitive. The first character cannot be an asterisk (\*), and the first three characters cannot be the letters IBM.

The defaults are ADSMSERV1, ADSMSERV2, ADSMSERV3, and ADSMSERV4 (depending on the LANAdapter statements you have). For example, if you have LANAdapter 0 and LANAdapter 1, the defaults are ADSMSERV1 and ADSMSERV2.

## Examples

```
NETBiosname ADSMSERV1
```



---

## NETBIOSsession

Specifies the NetBIOS session available on an ADSM server.

### Syntax

```
▶▶—NETBIOSsession—value————▶▶
```

### Parameters

*name*

Specifies the number of the NetBIOS sessions that are reserved for this process per adapter. The minimum number of sessions is 1 and the maximum number of sessions is 255.

This option is independent of the MAXSESSION option other than MAXSESSION is the default number of the NetBIOS session. The limit specified in the MAXSESSION option overrides the NETBIOSsession limit for allowing a scheduled session to start.

### Examples

```
NETBIOSsession 16
```

---

## NOAUDITStorage

As part of a license audit operation, the server calculates, by node, the amount of backup, archive, and space management storage in use. For servers managing large amounts of data, this calculation can take a great deal CPU time and can stall other server activity. You can use the NOAUDITSTORAGE option to specify that storage is not to be calculated as part of a license audit.

### Syntax

```
▶▶—NOAUDITStorage————▶▶
```

### Parameters

None

---

## NUMberformat

Specifies the format by which numbers are displayed by the server.

### Syntax

```
▶▶—NUMberformat—number————▶▶
```

## Parameters

*number*

Select a number from 1 to 6 to identify the number format used by the server.  
The default is 1.

1	1,000.00
2	1,000,00
3	1 000,00
4	1 000.00
5	1.000,00
6	1'000,00

## Examples

NUMberformat 4

---

## SHMQUEUENAME

Specifies the Shared Memory queue name used for the client/server handshake.

This communication method is ideal when running the server and client on the same OS/2 machine because Shared Memory is internal to the OS/2 base system and requires less system overhead than Named Pipes.

## Syntax

▶▶—SHMQUEUENAME—*qname*————▶▶

## Parameters

The default is \QUEUES\ADSM\DSMSERV. The Queue name must begin with \QUEUES.

## Examples

SHMQUEUENAME

---

## SNABuffersize

Specifies the size of the APPC send and receive buffers.

## Syntax

▶▶—SNABuffersize—*value*————▶▶

## Parameters

*value*

Specifies the size of the APPC send and receive buffers in kilobytes.

## Examples

```
SNABuffersize 15360
```

---

## STAtusmsgcnt

Specifies the number of records (times 1000) that will be processed between status messages during DSMSERV DUMPDB and DSMSERV LOADDDB commands.

## Syntax

```
▶▶—STAtusmsgcnt—nnn————▶▶
```

## Parameters

*nn*

Specifies a number between 1 and 10000 (this number is multiplied by 1000).

The default value is 10. The minimum value is 1.

---

## TCPNodeLAY

Specifies whether the server should send small amounts of data or allow TCP/IP to buffer the data. Disallowing buffering may improve throughput but more packets are sent over the network.

## Syntax

```
▶▶—TCPNodeLAY— Yes  No————▶▶
```

## Parameters

**Yes**

Specifies that TCP/IP should not buffer small ADSM messages. The default is Yes.

**No**

Specifies that TCP/IP should buffer small ADSM messages.

## Examples

```
TCPNodeLAY Yes
```

---

## TCPPort

Specifies the port number on which the server TCP/IP communication driver is to wait for requests.

### Syntax

```
▶▶—TCPPort—portnumber————▶▶
```

### Parameters

*portnumber*

Specifies the port address of the server. The default value is 1500.

### Examples

```
TCPPort      1500
```

---

## TCPWindowSize

Specifies the size you want to use for the TCP/IP sliding window for your client node. This is the size of the buffer used when sending or receiving data.

Larger window sizes may improve performance but uses more memory.

### Syntax

```
▶▶—TCPWindowSize—window_size————▶▶
```

### Parameters

*window\_size*

Specifies the size you want to use, in kilobytes, for the TCP/IP sliding window for your client node. You can specify a value from 0 to 2048. Values from 1 to 2048 indicate that the window size is in the range of 1K to 2M.

### Examples

```
TCPWindowSize 8KB
```

---

## TIMEformat

Specifies the format in which time is displayed by the server.

### Syntax

```
▶▶—TIMEformat—number————▶▶
```

## Parameters

*number*

Select a number from 1 to 4 to identify the time format used by the server. The default is 1.

1	23:00:00
2	23,00,00
3	23.00.00
4	12:00:00A/P

## Examples

TIMEformat 4

---

## TXNGroupmax

Specifies the number of files that are transferred as a group between a client and the server between transaction commit points. You can improve the performance of the backup-archive client BACKUP, ARCHIVE, RESTORE, and RETRIEVE commands by using a larger value for this option.

The minimum value is 4 files and the maximum value is 256 files. The default value is 16 files. The files transferred for the above commands are either actual files, directories, or both. ADSM counts each file and each directory as one file.

This option is related to the TXNBYTELIMIT option in the client options file. TXNBYTELIMIT controls the number of bytes, as opposed to files, that are transferred in a group of files between transaction commit points. At the completion of transferring a file, the client commits the transaction if the number of bytes transferred during the transaction reaches or exceeds the value of TXNBYTELIMIT, regardless of the number of files transferred.

## Syntax

```
▶▶—TXNGroupmax—numfiles————▶▶
```

## Parameters

*numfiles*

Specifies a number from 4 to 256 for the maximum number of files per transaction. The default is 16.

## Examples

TXNGroupmax 50

---

## VOLUMEHISTORY

Specifies files to be automatically updated whenever server sequential volume history information is changed.

A server options file that is created during installation includes a VOLUMEHISTORY option that specifies a default volume history file (volhist.out).

### Syntax

```
▶▶—VOLUMEHISTORY—name—————▶▶
```

### Parameters

*name*

Specifies the name of the files in which you want ADSM to store a backup copy of the volume history information collected by the server.

---

## Glossary

The terms in this glossary are defined as they pertain to the ADSM library. If you do not find the term you are looking for, refer to the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

This glossary may include terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York 10036.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC2/SC1).

## A

**absolute.** A backup copy group mode value indicating that a file is considered for incremental backup even if the file has not changed since the last backup. See also *mode*. Contrast with *modified*.

**access mode.** A storage pool and storage volume attribute that specifies whether data can be written to or read from storage pools or storage volumes. The access mode can be read/write, read-only, or unavailable. Volumes in primary storage pools can also have an access mode of destroyed. Volumes in copy storage pools can also have an access mode of offsite.

**accounting facility.** A facility that records statistics about client session activity.

**accounting records.** Files that record session resource usage at the end of each client session.

**action choice.** A choice in a pull-down menu that causes an action. See also *routing choice*.

**activate.** The process of validating the contents of a policy set and copying the policy set to the ACTIVE policy set.

**active policy set.** The policy set within a policy domain that contains the most recently activated policy currently in use by all client nodes assigned to that policy domain. See *policy set*.

**active version.** The most recent backup copy of a file stored by ADSM. Such a file is exempt from deletion until a backup detects that the user has either replaced the file with a newer version, or has explicitly deleted the file from the workstation. Contrast with *inactive version*.

**activity log.** A log that records normal activity messages generated by the server. These messages include information about server and client operations, such as the start time of sessions or device I/O errors. Each message includes a message ID, date and time stamp, and a text description. The number of days to retain messages in the activity log can be specified.

**administrative client.** A program that runs on a file server, workstation, or mainframe that allows administrators to control and monitor the server through administrator commands. Contrast with *backup-archive client*.

**administrative command schedule.** A database record that describes the planned processing of an administrative command during a specific time period. See also *client schedule*.

**administrative privilege class.** A permission granted to an administrator that controls the commands that the administrator can issue. See *system privilege class*, *analyst privilege class*, *operator privilege class*, *policy privilege class* or *storage privilege class*.

**administrative session.** A period of time in which an administrator user ID can communicate with a server to perform administrative tasks. Contrast with *client node session*.

**administrator.** A user who has been registered to the server. Administrators can be authorized to one or more of the following administrative privilege classes: system, policy, storage, operator, or analyst. Administrators can use the administrative client to enter server commands and queries in accordance with their privileges.

**administrator definition.** Server control information that includes the administrator's name, password, contact information, administrative privilege classes, policy domains and storage pools assigned to an administrator, and whether the administrative ID is locked from the server. An administrator definition can be exported from a source server and imported to a target server at a later date.

**ADSM.** ADSTAR Distributed Storage Manager.

**ADSM application program interface (API).** A set of functions that applications running on a client platform can call to store, query, and retrieve objects from ADSM storage.

**ADSTAR Distributed Storage Manager (ADSM).** A client/server program that provides storage management to customers in a multivendor computer environment.

**Advanced Interactive Executive (AIX).** An operating system used in the RISC System/6000 computers. The AIX operating system is IBM's implementation of the UNIX operating system.

**Advanced Peer-to-Peer Networking (APPN).** An extension to the LU6.2 peer orientation for end-user services. See *SNA LU6.2* and *Systems Network Architecture*.

**Advanced Program-to-Program Communication (APPC).** An implementation of the SNA/SDLC LU6.2 protocol that allows interconnected systems to communicate and share the processing of programs. See *SNA LU6.2*, *Systems Network Architecture*, and *Common Programming Interface Communications*.

**AFS.** Andrew file system.

**AIX.** Advanced Interactive Executive.

**analyst privilege class.** An administrative privilege class that allows an administrator to reset statistics.

**Andrew file system (AFS).** A distributed file system developed for UNIX operating systems.

**API.** Application program interface.

**APPC.** Advanced Program-to-Program Communication.

**APPN.** Advanced Peer-to-Peer Networking.

**archive.** A function that allows users to copy one or more files to a storage pool for long-term storage. Archive copies may be accompanied by descriptive information and may be retrieved by archive date, by file name, or by description. Contrast with *retrieve*.

**archive copy.** A user file that has been archived to an ADSM storage pool.

**archive copy group.** A policy object containing attributes that control the generation, destination, and expiration of archive files. An archive copy group belongs to a management class.

**ARCHIVEPOOL.** A disk storage pool defined by ADSM at installation. It can be the destination for client files that are archived to the server. See *storage pool*.

**archive retention grace period.** The number of days ADSM retains an archive copy when the server is unable to rebind the file to an appropriate management class.

**AS/400.** Application System/400.

**assigned capacity.** The portion of available space that can be used to store database or recovery log information. See also *available space*.

**association.** The relationship between a client node and a client schedule. An association identifies the name of a schedule, the name of the policy domain to which the schedule belongs, and the name of a client node that performs scheduled operations.

**audit.** The process of checking for logical inconsistencies between information that the server has and the actual condition of the system. ADSM has processes for auditing volumes, the database, libraries, and licenses. For example, in auditing a volume ADSM checks for inconsistencies between information about backed up or archived files stored in the database and actual data associated with each backup version or archive copy in server storage.

**authentication.** The process of checking a user's password before allowing that user access to the server. Authentication can be turned on or off by an administrator with system privilege.

**autochanger.** A small multislot tape device that has a mechanism that automatically puts tape cartridges into the tape drive or drives. Also called *medium* or *media changer*, or a *library*.

**available space.** The amount of space, in megabytes, that is available to the database and recovery log. This space can be used to extend the capacity of the database or recovery log, or to provide sufficient free space before a volume is deleted from the database or recovery log.



## B

**background process.** A server process that runs in the background, allowing the administrative client to be used for other work.

**backup.** The process of copying information for safekeeping. ADSM has processes for backing up user files, the database, and storage pools. For example, users can back up one or more files to a storage pool to ensure against loss of data. Contrast with *restore*. See also *database backup series* and *incremental backup*.

**backup-archive client.** A program that runs on a workstation or file server and provides a means for users to back up, archive, restore, and retrieve files. Contrast with *administrative client*.

**backup copy.** A user file that has been backed up to an ADSM storage pool.

**backup copy group.** A policy object containing attributes that control the generation, destination, and expiration of backup files. A backup copy group belongs to a management class.

**BACKUPPOOL.** A disk storage pool defined by ADSM at installation. It can be the destination for client files that are backed up to the server. See *storage pool*.

**backup retention grace period.** The number of days ADSM retains a backup version after the server is unable to rebind the file to an appropriate management class.

**backup series.** See *database backup series*.

**backup version.** A file, directory, or file space that a user has backed up, which resides in ADSM server storage. There may be more than one backup version of a file in the storage pool, but at most only one is an active backup version. See *active version* and *inactive version*.

**binding.** The process of associating a file with a management class name. See *rebinding*.

**buffer.** Storage used to compensate for differences in the data rate flow, when transferring data from one device to another.

**buffer pool.** Temporary space used by the server to hold database or recovery log pages. See *database buffer pool* and *recovery log buffer pool*.

**buffer pool size.** The size of an area in memory used to store database or recovery log pages.

**bus converter.** A device that translates between different Hewlett-Packard internal I/O bus architectures.

## C

**cache.** The process of leaving a duplicate copy on random access media when the server migrates a file to another storage pool in the hierarchy.

**cartridge.** A sequential storage media that contains magnetic tape in a protective housing. Contrast with *tape reel*.

**CARTRIDGE.** On ADSM servers that support it, a device class that is used to categorize tape devices that support tape cartridges, such as the 3495 Tape Library Dataserver.

**cartridge system tape (CST).** The base tape cartridge media used with 3480 or 3490 Magnetic Tape Subsystems. When specified as a media type in ADSM, CST identifies standard length tape. Contrast with *enhanced capacity cartridge system tape*.

**central scheduler.** A function that allows an administrator to schedule client operations and administrative commands. The operations can be scheduled to occur periodically or on an explicit date. See *client schedule* and *administrative command schedule*.

**CID.** Configuration Installation and Distribution.

**client.** A program running on a PC, workstation, file server, LAN server, or mainframe that requests services of another program, called the server. There are three types of ADSM clients: administrative, backup-archive, and space management. See *administrative client*, *backup-archive client*, and *space management client*.

**Client Access/400.** A software product that supports advanced program-to-program communications (APPC) in the DOS, OS/2, and Microsoft Windows environments and provides a set of end user services.

**client domain.** The set of drives, file systems, or volumes selected by a backup-archive client user during a backup or archive operation.

**client migration.** The process of copying a file from a client node to ADSM storage and replacing the file with

a stub file on the client node. The process is controlled by the user and by space management attributes in the management class. See also *space management*.

**client node.** A file server or workstation on which the backup-archive client program has been installed, which has been registered to the server.

**client node definition.** Server control information that includes the client's user ID, password, contact information, policy domain, file compression status, deletion authority, and whether the user ID is locked from the server. A client node definition can be exported from a source server so that it can be imported to a target server at a later date.

**client node session.** A period of time in which a user can communicate with a server to perform backup, archive, restore, or retrieval requests. Contrast with *administrative session*.

**client options file.** A file that a client can edit, containing a default set of processing options that identify the server, communication method, backup and archive options, space management options, and scheduling options. Also called the *dsm.opt* file.

**client polling scheduling mode.** A client/server communication technique where the client queries the server for work.

**client schedule.** A database record that describes the planned processing of a client operation during a specific time period. The client operation can be a backup, archive, restore, or retrieve operation, a client operating system command, or a macro. See also *administrative command schedule*.

**client/server.** A system architecture in which one or more programs (clients) request computing or data services from another program (server).

**client system options file.** A file, used on UNIX clients, containing a default set of processing options that identify the ADSM servers to be contacted for services. This file also specifies communication methods and options for backup, archive, space management, and scheduling. Also called the *dsm.sys* file. See also *client user options file*.

**client user options file.** A user-created file, used on UNIX clients, containing a default set of processing options that identify the server, communication method, backup and archive options, space management options,

and scheduling options. Also called the *dsm.opt* file. See also *client system options file*.

**closed registration.** A registration process in which an administrator must register workstations as client nodes with the server. Contrast with *open registration*.

**collocation.** A process that attempts to keep all data belonging to a single client node on a minimal number of sequential access media volumes within a storage pool. The purpose of collocation is to minimize the number of volumes that must be accessed when a large amount of data must be restored.

**command line interface.** A type of user interface where commands are specified on the command line when the backup-archive or administrative client is started. Contrast with *graphical user interface*.

**commit.** To make changes permanent in the database files. Changes made to the database files are not permanent until they are committed.

**Common Programming Interface Communications (CPI-C).** A programming interface that allows program-to-program communication using SNA LU6.2. See also *Systems Network Architecture*.

**Common User Access (CUA).** Guidelines for the dialog between a human and a workstation or terminal. One of the three SAA architectural areas.

**communication manager.** A component of OS/2 that allows a workstation to connect to a host computer and use the host resources as well as the resources of other personal computers to which the workstation is attached, either directly or through a host.

**communication method.** The method used by a client and server for exchanging information.

**communication protocol.** A set of defined interfaces that allow computers to communicate with each other.

**compression.** The process of saving storage space by eliminating empty fields or unnecessary data to shorten the length of the file. In ADSM, compression can occur at a workstation before files are backed up or archived to server storage. On some types of tape drives, hardware compression can be used.

**Configuration Installation and Distribution (CID).** IBM's term for capabilities to automate installation. CID-enabled products are capable of unattended, remote installation.

**contextual help.** A type of online help that provides specific information for each selectable object, menu choice, notebook tab, field, and control or push button in a window.

**conversion.** On VM servers, the process of changing from WDSF/VM to ADSM.

**copy group.** A policy object that contains attributes that control the generation, destination, and expiration of backup and archive files. There are two kinds of copy groups: backup and archive. Copy groups belong to management classes. See also *frequency*, *destination*, *mode*, *serialization*, *retention*, and *version*.

**copy status.** The status of volume copies defined to the database or recovery log. The copy status can be synchronized, stale, off-line, or undefined.

**copy storage pool.** A named set of volumes that contains copies of files that reside in primary storage pools. Copy storage pools are used to back up the data stored in primary storage pools. A copy storage pool cannot be a destination for a backup copy group, an archive copy group, or a management class (for space-managed files). See *primary storage pool* and *destination*.

**CPI-C.** Common Programming Interface Communications.

**CST.** Cartridge system tape.

**CUA.** Common User Access.

## D

**daemon.** In the AIX operating system, a program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their tasks; others operate periodically.

**daemon process.** In the AIX operating system, a process begun by the root user or by the root shell that can be stopped only by the root user. Daemon processes generally provide services that must be available at all times, such as sending data to a printer.

**damaged file.** A file for which ADSM has detected data-integrity errors.

**DASD.** Direct access storage device.

**database.** A collection of information about all objects managed by the server, including policy management objects, users and administrators, and client nodes.

**database audit.** A utility that checks for and optionally corrects inconsistent database references.

**database backup series.** One full backup of the database, plus up to 32 incremental backups made since that full backup. Each full backup that is run starts a new database backup series. A backup series is identified with a number.

**database backup trigger.** A set of criteria that defines when and how database backups are run automatically. The criteria determine how often the backup is run, whether the backup is a full or incremental backup, and where the backup is stored.

**database buffer pool.** Storage that is used as a cache to allow database pages to remain in memory for long periods of time, so that the server can make continuous updates to pages without requiring input or output (I/O) operations from external storage.

**database dump.** A utility that copies database entries to media for later reload in case a catastrophic error should occur.

**database load.** A utility that copies database entries from media to a newly installed database.

**database volume.** A volume that has been assigned to the database.

**dataserver.** See *Tape Library Dataserver*.

**data set.** See *linear data set*.

**data storage.** The primary and copy storage pools used by the server to store users' files: backup versions, archive copies, and files migrated from client nodes (space-managed files). Synonymous with *server storage*. See *primary storage pool*, *copy storage pool*, *storage pool volume*, and *volume*.

**DDM.** Distributed Data Management.

**default management class.** A management class assigned to a policy set, which is used to govern backed up or archived files when a user does not explicitly bind a file to a specific management class.

**definition.** Server control information that includes administrator, client node, and policy definitions. A definition can be exported from a source server to

external media so that it can be imported to a target server at a later date.

**deletion exit.** An installation-wide exit that informs a tape management system or operator that the server has deleted a sequential access media volume from its database.

**delimiter.** (1) A character used to indicate the beginning and end of a character string. (2) A character that groups or separates words or values in a line of input.

**density.** On MVS and VM servers, a device class attribute that identifies the bits per inch that can be stored on tape reels. ADSM supports 1600 and 6250 bits per inch (bpi).

**desktop.** On-screen representation of a desk top.

**desktop client.** The group of clients supported by ADSM that are not UNIX-based and are not OpenEdition MVS. For example, a DOS client is a desktop client.

**destination.** A copy group or management class attribute that specifies the primary storage pool to which a client file will be backed up, archived, or migrated. At installation, ADSM provides storage destinations named BACKUPPOOL, ARCHIVEPOOL, and SPACEMGPOOL.

**device class.** A named group of storage devices. Each device class has a unique name and represents a device type of disk, file, optical disk, or tape.

**device configuration file.** A file that contains information about defined device classes, and, on AIX servers, defined libraries and drives. The file can be created by using an ADSM command or by using an option in the server options file. The information is a copy of the device configuration information in the ADSM database.

**device driver.** A collection of subroutines that control the interface between I/O device adapters and the processor.

**device type.** A category of storage device. Each device class must be categorized with one of the supported device types, for example, DISK or CARTRIDGE.

**direct access storage device (DASD).** A device in which access time is effectively independent of the location of the data.

**DISK.** A device class that is defined by ADSM at installation. It is used to categorize disk drives, such as 3390 DASD or 3380 DASD.

**diskette.** A small, magnetic disk enclosed in a jacket.

**disk operating system (DOS).** An operating system used in IBM PC, PS/2, and compatible computers.

**Distributed Data Management (DDM).** A feature of the System Support Program Product that allows an application program (client) to use server program functions to work on files that reside in a remote system.

**DLL.** Dynamic link library.

**DLT.** Digital linear tape.

**domain.** See *policy domain* or *client domain*.

**DOS.** Disk operating system.

**drive.** A device used to read and write data on a medium such as a disk, diskette, or tape.

**dsm.opt file.** See *client options file* and *client user options file*.

**dsm.serv.opt.** See *server options file*.

**dsm.sys file.** See *client system options file*.

**dynamic.** A copy group serialization value that specifies that ADSM accepts the first attempt to back up or archive a file regardless of whether the file is modified during the backup or archive process. See also *serialization*. Contrast with *shared dynamic*, *shared static*, and *static*.

**dynamic link library.** A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

## E

**ECCST.** Enhanced capacity cartridge system tape.

**enhanced capacity cartridge system tape (ECCST).** Cartridge system tape with increased capacity that can only be used with 3490E tape subsystems. Contrast with *cartridge system tape*.

**error log.** A character file written on random access media that contains information about errors detected by the server or client.

**estimated capacity.** The available space, in megabytes, of a storage pool.

**Ethernet.** A data link protocol and LAN that interconnects personal computers and workstations via coaxial cable.

**event.** Administrative commands or client operations that are scheduled to be executed at a particular time.

**event record.** A database record that describes actual status and results for events.

**exclude.** The process of identifying files or directories in an include-exclude list to prevent these objects from being backed up whenever a user or schedule issues an incremental or selective backup operation, or to prevent these objects from being migrated off the client node via ADSM space management.

**exclude-include list.** See *include-exclude list*.

**exit.** To execute an instruction within a portion of a computer program in order to terminate the execution of that portion.

**exit machine.** On a VM server, a virtual machine that runs the mount and deletion installation-wide exits on VM systems.

**expiration.** The process by which files are identified for deletion because their expiration date or retention period has passed. Backed up or archived files are marked expired by ADSM based on the criteria defined in the backup or archive copy group.

**expiration date.** On MVS, VM, and VSE servers, a device class attribute used to notify tape management systems of the date when ADSM no longer needs a tape volume. The date is placed in the tape label so that the tape management system does not overwrite the information on the tape volume before the expiration date.

**export.** The process of copying administrator definitions, client node definitions, policy definitions, server control information or file data to external media.

**export/import facility.** See *import/export facility*.

**extend.** The process of increasing the portion of available space that can be used to store database or recovery log information. Contrast with *reduce*.

## F

**file data.** File space definitions, authorization rules, backed up files, archive copies, and space-managed files. File data can be exported from a source server to external media so that it can be imported to a target server at a later date.

**file record extent.** The extent of the file enumerated in number of records.

**file space.** A logical space in a client's storage that can contain a group of files. For clients on systems such as OS/2, a file space is a logical partition and is identified by a volume label. For clients on systems such as AIX and UNIX, a file space can consist of any subset of directories and subdirectories stemming from a virtual mount point. Clients can restore, retrieve, or delete their file spaces from ADSM server storage. ADSM does not necessarily store all the files from a single file space together, but can identify all the files in server storage that came from a single file space.

**File Transfer Protocol (FTP).** In TCP/IP, the protocol that makes it possible to transfer data among hosts and to use foreign hosts indirectly.

**format.** A device class attribute that specifies the recording format used to read or write to sequential access media, for example to cartridge tape.

**frequency.** A copy group attribute that specifies the minimum interval, in days, between incremental backups.

**FTP.** File Transfer Protocol.

**full backup.** An ADSM function that copies the entire database. A full backup begins a new database backup series. Contrast with *incremental backup*. See *database backup series*.

**fuzzy copy.** A backup version or archive copy of a file that might not accurately reflect what is currently in the file because ADSM backed up or archived the file while the file was being modified.

## G

**general help.** A type of online help that provides an overview of the function of the window.

**graphical user interface (GUI).** A type of user interface that takes advantage of a high-resolution monitor, including some combination of graphics, the object-action paradigm, the use of pointing devices, menu bars, overlapping windows, and icons. See *windowed interface*. Contrast with *command line interface*.

**group of mirrored volumes.** One, two, or three volume copies defined to the database or recovery log. Each volume copy in the group contains exactly the same portion of the database or recovery log. See *mirroring*.

**GUI.** Graphical user interface.

## H

**handle.** A data structure that is a temporary local identifier for an object. A handle identifies an object at a specific location by binding it.

**HDA.** Head-disk assembly.

**head-disk assembly (HDA).** A field replaceable unit in a direct access storage device containing the disks and actuators.

**help index.** A type of online help that provides an alphabetic listing of all help topics.

**hierarchical storage management (HSM) client.** A program that runs on workstations to allow users to maintain free space on their workstations by migrating and recalling files to and from ADSM storage. The HSM client allows use of ADSM space management functions. Synonymous with *space management client*.

**high migration threshold.** A percentage of the storage pool capacity that identifies when ADSM can start migrating files to the next available storage pool in the hierarchy. Contrast with *low migration threshold*. See *server migration*.

**HP-UX.** Hewlett-Packard UNIX operating system. HP-UX is one of the operating systems that ADSM supports as a client environment and a server environment.

**HSM client.** Hierarchical storage management client.

## I

**import.** The process of copying administrator definitions, client node definitions, policy definitions, server control information or file data from external media to a target server.

**import/export facility.** The facility that allows system administrators to copy definitions and file data from a source server to external media to move or copy information between servers. Any subset of information can be imported to a target server from the external media.

**inactive version.** A backup version of a file for which a more recently backed up version exists. Inactive backup versions are eligible for expiration processing according to the management class assigned to the file. Contrast with *active version*.

**include-exclude file.** On UNIX clients, a file containing statements that ADSM uses to determine whether to back up or migrate certain files, and to determine the associated management classes to use for backup, archive, and space management. See *include-exclude list*.

**include-exclude list.** A group of include and exclude option statements in a file. ADSM uses the statements to determine whether to back up or migrate certain files, and to determine the associated management classes to use for backup, archive, and space management. The exclude options identify files that should not be backed up or migrated off the client node. The include options identify files that are exempt from the exclusion rules, or assign a management class to a file or group of files for backup, archive, or space management services. The include-exclude list is defined either in the include-exclude file (for UNIX clients) or in the client options file (for other clients).

**inconsistencies.** Any discrepancy between the information recorded in the database about backed up or archived files and the actual data associated with backed up or archived files residing in server storage.

**incremental backup.** (1) A function that allows users to back up files or directories that are new or have changed since the last incremental backup. With this function, users can back up files or directories from a client domain that are not excluded in the include-exclude list and that meet the requirements for

frequency, mode, and serialization as defined in the backup copy group of the management class assigned to the files. Contrast with *selective backup*. (2) An ADSM function that copies only the pages in the database that are new or changed since the last full or incremental backup. Contrast with *full backup*. See *database backup series*.

**internal mounting facility.** On a VM server, a VM facility that allows the server to request tape mounts by sending a message to a mount operator. The message is repeated until the tape is mounted or until the mount wait time is exceeded.

**inter-user communication vehicle (IUCV) facility.** On a VM server, a VM communication method used to pass data between virtual machines and VM components.

**IPX/SPX.** Internetwork Packet Exchange/Sequenced Packet Exchange. IPX/SPX is Novell NetWare's communication protocol.

**IUCV.** Inter-user communication vehicle.

## K

**KB.** Kilobyte.

**kernel.** The part of an operating system that performs basic functions such as allocating hardware resources.

**kernel extension.** A program that modifies parts of the kernel that can be customized to provide additional services and calls. See *kernel*.

**kilobyte (KB).** 1024 bytes.

## L

**LAN.** Local area network.

**length.** A device class attribute that specifies the length of cartridge tape by specifying one of the following media types: CST for standard length tape or ECCST for double length tape.

**library.** (1) A repository for demountable recorded media, such as magnetic tapes. (2) In ADSM, a collection of one or more drives, and possibly robotic devices (depending on the library type), which can be used to access storage volumes. (3) In the AS/400 system, a system object that serves as a directory to

other objects. A library groups related objects, and allows the user to find objects by name.

**linear data set.** A type of MVS data set that ADSM uses for the database, the recovery log, and storage pools. The data set must be preallocated using VSAM IDCAMS and formatted by ADSM for its use. See *minidisk*.

**load.** See *mount*.

**local area network (LAN).** A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

**log pool size.** The size of an area in memory used to store recovery log pages.

**logical volume.** The combined space from all volumes defined to either the database or the recovery log. In ADSM, the database is one logical volume and the recovery log is one logical volume.

**low migration threshold.** A percentage of the storage pool capacity that specifies when ADSM can stop the migration of files to the next storage pool. Contrast with *high migration threshold*. See *server migration*.

## M

**macro file.** An optional file that contains one or more administrative commands and is invoked from an administrative client.

**management class.** A policy object that users can bind to each file to specify how the server manages the file. The management class can contain a backup copy group, an archive copy group, and space management attributes. The copy groups determine how the ADSM server manages backup versions or archive copies of files. The space management attributes determine whether files are eligible for migration from client nodes to ADSM storage, and under what conditions. See also *copy group*, *binding* and *rebinding*.

**mask.** A pattern of characters that controls the keeping, deleting, or testing of positions of another pattern of characters or bits.

**maximum extension.** Specifies the maximum amount of storage space, in megabytes, that you can extend the database or recovery log.

**maximum reduction.** Specifies the maximum amount of storage space, in megabytes, that you can reduce the database or recovery log.

**maximum utilization.** The highest percentage of assigned capacity used by the database or recovery log.

**MB.** Megabyte.

**megabyte (MB).** (1) For processor storage and real and virtual memory, 2<sup>20</sup> or 1 048 576 bytes. (2) For disk storage capacity and transmission rates, 1 000 000 bytes.

**migrate.** (1) To move data from one storage pool to the storage pool specified as the next pool in the hierarchy. The process is controlled by the high and low migration thresholds for the first storage pool. See *high migration threshold* and *low migration threshold*. (2) To copy a file from a client node to ADSM storage. ADSM replaces the file with a stub file on the client node. The process is controlled by the include-exclude list and by space management attributes in management classes.

**migration.** The process of moving data from one storage location to another. See *client migration* and *server migration*.

**minidisk.** A logical subdivision of a VM physical disk that provides storage on contiguous cylinders of DASD. On a VM server, a minidisk can be defined as a disk volume that can be used by the database, recovery log, or a storage pool. See also *linear data set*.

**mirroring.** A feature that protects against data loss within the database or recovery log by writing the same data to multiple disks at the same time. Mirroring supports up to three exact copies of each database or recovery log volume. See *group of mirrored volumes*.

**mm.** Millimeter.

**mode.** A copy group attribute that specifies whether to back up a file that has not been modified since the last time the file was backed up. See *modified* and *absolute*.

**modified.** A backup copy group mode value indicating that a file is considered for incremental backup only if it has changed since the last backup. A file is considered changed if the date, size, owner, or permissions have changed. See *mode*. Contrast with *absolute*.

**Motif.** A graphical user interface that performs window management and contains a high level toolkit for application program development. It provides an icon view of the UNIX file system. Also known as X-Windows/Motif or Motif X—Toolkit.

**mount.** To place a data medium (such as a tape cartridge) on a drive in a position to operate.

**mount exit.** On a VM server, an installation-wide exit (DSMMOUNT EXEC) that requests tape mounts on behalf of the server on VM systems.

**mount limit.** A device class attribute specifying the maximum number of volumes that can be simultaneously accessed from the same device class, that is, the maximum number of mount points. See *mount point*.

**mount operator.** On a VM server, a VM user ID that can receive tape mount messages from the server.

**mount point.** A logical drive through which ADSM accesses volumes in a sequential access device class. For a device class with a removable media device type (for example, CARTRIDGE), a mount point is a logical drive associated with a physical drive. For a device class with the device type of FILE, a mount point is a logical drive associated with an I/O stream. The number of mount points for a device class is determined by the mount limit for that class. See *mount limit*.

**mount request.** A server request to mount a sequential access media volume so that data can be read from or written to the sequential access media.

**mount retention period.** A device class attribute that specifies the maximum amount of time, in minutes, that the server retains a mounted sequential access media volume that is not being used before it dismounts the sequential access media volume.

**mount wait period.** A device class attribute that specifies the maximum amount of time, in minutes, that the server waits for a sequential access volume mount request to be satisfied before canceling the request.

**Multiple Virtual Storage (MVS).** One of the family of IBM operating systems for the System/370 or System/390 processor, such as MVS/ESA. MVS is one of the supported server environments.

**MVS.** Multiple Virtual Storage.



## N

**Named Pipes.** A communication protocol that is built into the OS/2 operating system. It can be used to establish communications between an ADSM/2 server and OS/2 clients. The client and ADSM/2 server must reside on the same system.

**NetBIOS.** Network Basic Input/Output System.

**network adapter.** A physical device, and its associated software, that enables a processor or controller to be connected to a network.

**Network Basic Input/Output System (NetBIOS).** An operating system interface for application programs used on IBM personal computers that are attached to the IBM Token-Ring Network.

**Network File System (NFS).** A protocol defined by Sun Microsystems that extends TCP/IP network file services. NFS permits remote node files to appear as though they are stored on a local workstation.

**Networking Services/DOS (NS/DOS).** A software product that supports advanced program-to-program communications (APPC) in the DOS and Microsoft Windows 3.1 environments. With NS/DOS, communications applications on your workstation "talk to" partner applications on other systems that support APPC.

**NFS.** Network File System.

**node.** A unique name used to identify a workstation to the server. See also *client node*.

**notebook.** A graphical representation that resembles a spiral-bound notebook that contains pages separated into sections by tabbed divider-pages. A user can "turn" the pages of a notebook to move from one section to another.

**notify operator.** A VM user ID that specifies an operator who receives messages about severe errors and abnormal conditions.

## O

**object.** A collection of data managed as a single entity.

**offsite volume.** A removable media volume that is at a location where it cannot be mounted for use.

**OpenEdition MVS.** MVS/ESA services that support an environment within which operating systems, servers, distributed systems, and workstations share common interfaces. OpenEdition MVS supports standard application development across multivendor systems and is required to create and use applications that conform to the POSIX standard.

**open registration.** A registration process in which users can register their own workstations as client nodes with the server. Contrast with *closed registration*.

**Operating System/2 (OS/2).** An operating system used in IBM PC AT, PS/2, and compatible computers. OS/2 is one of the supported client environments and one of the supported server environments.

**operator privilege class.** An administrative privilege class that allows an administrator to issue commands that control the operation of the server. This privilege class allows disabling or halting the server to perform maintenance, enabling the server, canceling server processes, and managing tape.

**optical disk.** A disk that contains data readable by optical techniques.

**optical drive.** A drive mechanism that rotates an optical disc.

**optical library.** A disk storage device that houses optical disk drives and optical disks, and contains a mechanism for moving optical disks between a storage area and optical disk drives.

**OS/2.** Operating System/2.

**OS/400.** Operating System/400.

**owner.** The owner of backup-archive files sent from a multiuser client node, such as AIX.

## P

**page.** (1) A block of instructions, data, or both. (2) In ADSM, a unit of space allocation within database volumes. (3) In a virtual storage system, a fixed block that has a virtual address and is transferred as a unit between real storage and auxiliary storage.

**paging.** (1) The action of transferring instructions, data, or both, between real storage and external page storage. (2) Moving data between memory and a mass storage device as the data is needed.

**pattern-matching expression.** A string expression that uses wildcard characters to specify one or more ADSM objects. See also *wildcard character*.

**PC Support/400.** A software product that supports advanced program-to-program communications (APPC) in the DOS, OS/2, and Microsoft Windows environments and provides a set of end user services.

**platform.** The operating system environment in which a program runs.

**policy definition.** Server control information that includes information about policy domains, policy sets (including the ACTIVE policy set), management classes (including the default management class), copy groups, schedules, and associations between client nodes and schedules. A policy definition can be exported from a source server so that it can be imported to a target server at a later date.

**policy domain.** A policy object that contains policy sets, management classes, and copy groups that is used by a group of client nodes. See *policy set*, *management class*, and *copy group*.

**policy privilege class.** An administrative privilege class that allows an administrator to manage policy objects, register client nodes, and schedule client operations (such as backup services) for client nodes. Administrators can be authorized with unrestricted or restricted policy privilege. See *unrestricted policy privilege* or *restricted policy privilege*.

**policy set.** A policy object that contains a group of management class definitions that exist for a policy domain. At any one time there can be many policy sets within a policy domain but only one policy set can be active. See *management class* and *active policy set*.

**premigration.** For an HSM client, the process of copying files that are eligible for migration to ADSM storage, but leaving the original file intact on the local system.

**primary storage pool.** A named set of volumes that ADSM uses to store backup versions of files, archive copies of files, and files migrated from client nodes via ADSM space management. A primary storage pool may be backed up to a copy storage pool either automatically or by command. See *destination* and *copy storage pool*.

**privilege class.** A level of authority granted to an ADSM administrator. ADSM has five privilege classes: system, policy, storage, operator, and analyst. The privilege class determines which ADSM administrative tasks the administrator can perform. For example, an administrator with system privilege class can perform any administrative task.

**programmable workstation communication services (PWSCS).** A product that provides transparent high performance communications between programs running on workstations or on host systems.

**protection status.** A device class attribute that specifies whether to update the RACF profile to identify which users have access to cartridge tapes associated with this device class on MVS servers.

**PWSCS.** Programmable workstation communication services.

## Q

**QIC.** Quarter-inch cartridge (a type of magnetic tape media).

## R

**random access media.** Any volume accessed in a nonsequential manner. In ADSM, volumes are accessed in a nonsequential manner if they reside in the DISK device class.

**randomization.** The percentage of the startup window that the server can use to randomize start times for different client nodes associated with a schedule.

**rebinding.** The process of associating a file with a new management class name. For example, rebinding occurs when the management class associated with a file is deleted. See *binding*.

**recall.** A function that allows users to access files that have been migrated from their workstations to ADSM storage via ADSM space management. Contrast with *migrate*.

**reclamation.** A process of consolidating the remaining data from many sequential access media onto a single new sequential access media.

**reclamation threshold.** A value that specifies a percentage of space on sequential access media volumes that can be occupied by reclaimable space. The remainder of the space is for active data. (Space becomes reclaimable when files are expired.)

**recovery log.** A log of updates that are about to be written to the database. The log can be used to recover from system and media failures.

**recovery log buffer pool.** Used to hold new transactions records until they can be written to the recovery log.

**reduce.** The process of freeing up enough space to allow you to delete a volume from the database or recovery log. Contrast with *extend*.

**REEL.** On ADSM servers that support it, a device class that is used to categorize tape devices that support tape reels, such as the 3420 9-track tape device.

**register.** Defines a client node or administrator who can access the server. See *registration*.

**registration.** The process of identifying a client node or administrator to the server.

**reply operator.** On a VM server, a VM user ID that specifies an operator who will reply to tape mount requests by the server.

**restore.** The process of returning a backup copy to an active storage location for use. ADSM has processes for restoring its database, storage pools, storage pool volumes, and users' backed-up files. For example, users can copy a backup version of a file from the storage pool to the workstation. The backup version in the storage pool is not affected. Contrast with *backup*.

**restricted policy privilege.** An administrative privilege class that enables an administrator to manage policy objects only for the policy domains for which the administrator has been authorized.

**restricted storage privilege.** An administrative privilege class that enables an administrator to control the allocation and use of storage resources only for the storage pools for which the administrator has been authorized.

**retention.** The amount of time, in days, that inactive backed up or archived files will be retained in the storage pool before they are deleted. The following copy group attributes define retention: retain extra versions, retain only version, retain version.

**retention period.** On an MVS server, a device class attribute that specifies how long files are retained on sequential access media. When used, ADSM passes this information to the MVS operating system to ensure that other tape management systems do not overwrite tape volumes that contain retained data.

**retrieve.** A function that allows users to copy an archive copy from the storage pool to the workstation. The archive copy in the storage pool is not affected. Contrast with *archive*.

**RLIO.** Record Level Input/Output.

**rollback.** To remove changes that were made to database files since the last commit point.

**root.** In the AIX and UNIX environments, the user name for the system user with the most authority.

**root user.** In the AIX and UNIX environments, an expert user who can log in and execute restricted commands, shut down the system, and edit or delete protected files. Also called the *superuser*.

**routing choice.** A choice in a pull-down menu that, when selected, brings up another window. See also *action choice*.

## S

**SAA.** Systems Application Architecture.

**schedule.** A database record that describes scheduled client operations or administrative commands. See *administrative command schedule* and *client schedule*.

**scheduling mode.** The type of scheduling operation set for the server and client. ADSM supports two scheduling modes for client operations: client-polling and server-prompted.

**scratch volume.** A volume that is available for ADSM use. The volume is labeled, is either blank or contains no valid data, and is not defined to ADSM.

**SCSI.** Small computer system interface.

**selective backup.** A function that allows users to back up specific files or directories from a client domain. With this function, users can back up files or directories that are not excluded in the include-exclude list and that meet the requirement for serialization as defined in the backup copy group of the management class assigned to the files. Contrast with *incremental backup*.

**sequential access media.** Any volume that is accessed in a sequential manner, as opposed to a random manner. In ADSM, volumes are accessed sequentially if they reside in a device class other than DISK.

**serialization.** A copy group attribute that specifies what ADSM does if files are modified during back up or archive processing. The value of this attribute determines whether processing continues, is retried, or is stopped. See *static*, *dynamic*, *shared static*, and *shared dynamic*.

**server.** A program that provides services to other programs (clients).

**server migration.** The process of moving data from one storage pool to the next storage pool as controlled by the high and low migration thresholds. See *high migration threshold* and *low migration threshold*.

**server options file.** A file that specifies processing options for communication methods, tape handling, pool sizes, language, and date, time, and number formats.

**server program.** The program that provides backup, archive, space management, and administrative services to clients. The server program must be at the necessary level to provide all of these services.

**server-prompted scheduling mode.** A client/server communication technique where the server contacts the client when work needs to be done.

**server storage.** The primary and copy storage pools used by the server to store users' files: backup versions, archive copies, and files migrated from client nodes (space-managed files). Synonymous with *data storage*. See *primary storage pool*, *copy storage pool*, *storage pool volume*, and *volume*.

**session resource usage.** The amount of wait time, CPU time, and space used or retrieved during a client session.

**shared dynamic.** A copy group serialization value that specifies that a file must not be modified during a backup or archive operation. ADSM attempts to retry the backup or archive operation a number of times; if the file is in use during each attempt, ADSM will back up or archive the file on its last try even though the file is in use. See also *serialization*. Contrast with *dynamic*, *shared static*, and *static*.

**shared static.** A copy group serialization value that specifies that the file must not be modified during backup or archive. ADSM will retry the backup or archive operation a number of times; if the file is in use during each attempt, ADSM will not back up or archive the file. See also *serialization*. Contrast with *dynamic*, *shared dynamic*, and *static*.

**shell.** In the AIX and UNIX environments, a software interface between a user and the operating system of a computer. Shell programs interpret commands and user interactions on devices such as keyboards, pointing devices, and touch-sensitive screens and communicate them to the operating system.

**signal.** (1) A simple method of communication between two processes. One process can inform the other process when an event occurs. (2) In operating system operations, a method of inter-process communication that simulates software interrupts.

**signal handler.** A subroutine called when a signal occurs.

**SMIT.** System Management Interface Tool.

**SNA LU6.2.** Systems Network Architecture Logical Unit 6.2.

**socket.** (1) An endpoint for communication between processes or applications. (2) A pair consisting of TCP port and IP address, or UDP port and IP address.

**space-managed file.** A file that is migrated from and recalled to a client node via ADSM space management.

**space management.** The process of keeping sufficient free storage space available on a client node by migrating files to ADSM storage. The files are migrated based on criteria defined in management classes to which files are bound, and the include-exclude list.

Synonymous with *hierarchical storage management*. See also *migration*.

**space management client.** A program that runs on workstations to allow users to maintain free space on their workstations by migrating and recalling files to and from ADSM storage. Synonymous with *hierarchical storage management client*.

**SPACEMGPPOOL.** A disk storage pool defined by ADSM at installation. It can be the destination for files that are migrated from client nodes via ADSM space management. See *storage pool*.

**stale copy status.** Specifies that a volume copy is not available to the database or recovery log.

**STANDARD copy group.** A backup or archive copy group that is defined by ADSM at installation. See *copy group*.

**STANDARD management class.** A management class that is defined by ADSM at installation. See *management class*.

**STANDARD policy domain.** A policy domain that is defined by ADSM at installation. See *policy domain*.

**STANDARD policy set.** A policy set that is defined by ADSM at installation. See *policy set*.

**stanza.** A group of lines in a file that together have a common function or define a part of the system. Stanzas are usually separated by blank lines or colons, and each stanza has a name.

**startup window.** A time period during which a schedule must be initiated.

**static.** A copy group serialization value that specifies that the file must not be modified during backup or archive. If the file is modified during the attempt, ADSM will not back up or archive the file. See also *serialization*. Contrast with *dynamic*, *shared dynamic*, and *shared static*.

**storage hierarchy.** A logical ordering of primary storage pools, as defined by an administrator with system privilege. Generally, the ordering is based on the speed and capacity of the devices that the storage pools use. In ADSM, the storage hierarchy is defined by

identifying the *next* storage pool in a storage pool definition. See *storage pool*.

**storage management services.** A component that allows a central system to act as a file backup and archive server for local area network file servers and workstations.

**storage pool.** A named set of storage volumes that ADSM uses to store client data. A storage pool is either a primary storage pool or a copy storage pool. See *primary storage pool* and *copy storage pool*.

**storage pool volume.** A volume that has been assigned to an ADSM storage pool. See *volume*, *copy storage pool*, and *primary storage pool*.

**storage privilege class.** An administrative privilege class that allows an administrator to control the allocation and use of storage resources for the server, such as monitoring the database, recovery log, and server storage. Administrators can be authorized with unrestricted or restricted storage privilege. See *restricted storage privilege* or *unrestricted storage privilege*.

**stub file.** A file that replaces the original file on a client node when the file is migrated from the client node to ADSM storage.

**superuser.** See *root user*.

**synchronized copy status.** Specifies that the volume is the only volume copy or is synchronized with other volume copies in the database or recovery log. When synchronized, mirroring has started.

**system privilege class.** An administrative privilege class that allows an administrator to issue all server commands.

**Systems Application Architecture (SAA).** Software interfaces, conventions, and protocols that provide a framework for designing and developing applications that are consistent across systems.

**Systems Network Architecture (SNA).** A set of rules for data to be transmitted in a network. Application programs communicate with each other using a layer of SNA called advanced program-to-program communications (APPC).

## T

**tape.** A recording medium consisting of a long, narrow, flexible strip with a magnetic coating wound onto a reel or into a cartridge. See *cartridge* and *tape reel*.

**tape library.** (1) A term used to refer to a collection of tape cartridges. (2) An automated device that performs tape cartridge mounts and demounts without operator intervention.

**Tape Library Dataserver.** An automated tape library consisting of mechanical components, cartridge storage frames, IBM tape subsystems, and controlling hardware and software. The tape library dataserver performs tape cartridge mounts and demounts without operator intervention.

**tape reel.** A cylinder with flanges on which magnetic tape is wound. Devices such as the 3420 9-track tape device support tape reels. Contrast with *cartridge*.

**tape volume prefix.** A device class attribute that is the high-level-qualifier of the file name or the data set name in the standard tape label.

**task help.** A type of online help that provides a list of tasks that can be completed with a selected object. When you select a task, the help provides step-by-step information on how to complete the task.

**TCP/IP.** Transmission Control Protocol/Internet Protocol.

**Telnet.** In TCP/IP, the protocol that opens the connection to the system.

**Transmission Control Protocol/Internet Protocol (TCP/IP).** A set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks.

**trusted communication agent.** A program that performs communication tasks on behalf of the client or server, and ensures the security of the communications.

## U

**unit name.** On an MVS server, a device class attribute that specifies a group of tape devices used with the MVS server. A unit name can be a generic device type, an esoteric unit name, or a physical device.

**unrestricted policy privilege.** An administrative

privilege class that enables an administrator to manage policy objects for any policy domain.

**unrestricted storage privilege.** An administrative privilege class that enables an administrator to control the database, recovery log, and all storage pools.

**utilization.** The percent of assigned capacity used by the database or recovery log at a specific point of time.

## V

**validate.** The process of ensuring that the active policy set contains a default management class and reporting on copy group definition errors.

**version.** The maximum number of backup copies retained for files and directories. The following copy group attributes define version criteria: *versions data exists* and *versions data deleted*.

**Virtual Machine (VM).** One of the family of IBM operating systems for the System/370 or System/390 processor, including VM/ESA, VM/XA, VM/SP, and VM/HPO. VM is one of the supported server environments.

**Virtual Storage Extended (VSE).** One of the family of IBM operating systems for the System/370 or System/390 processor, including VSE/ESA. VSE is one of the supported server environments.

**VM.** Virtual Machine.

**volume.** The basic unit of storage for the database, recovery log, or a storage pool. A volume can be an LVM logical volume, a standard file system file, a tape cartridge, or an optical cartridge. Each volume is identified by a unique volume identifier. See *database volume*, *scratch volume*, and *storage pool volume*.

**volume history file.** A file that contains information about: volumes used for database backups and database dumps; volumes used for export of administrator, node, policy, or server data; and sequential access storage pool volumes that have been added, reused, or deleted. The information is a copy of the same types of volume information in the ADSM database.

**volume set.** An entire image of the database or recovery log, as displayed on the administrative graphical user interface.

**VSE.** Virtual Storage Extended.

## W

**WDSF/VM.** Workstation Data Save Facility/Virtual Machine.

**wildcard character.** A character or set of characters used to specify an unknown number or set of characters in a search string. Also called *pattern-matching character*.

**window.** A part of a display screen with visible boundaries in which information is displayed.

**windowed interface.** A type of user interface that is either a graphical user interface or a text based interface. The text based interface maintains a close affinity to the graphical user interface, including action

bars and their associated pull-downs and windows. See *graphical user interface*.

**workstation.** A personal computer system capable of maintaining data files.

**Workstation Data Save Facility/Virtual Machine (WDSF/VM).** The predecessor product to ADSTAR Distributed Storage Manager.

**WORM.** A type of optical media that can only be written to and cannot be erased.

## X

**X Windows.** A network transparent windowing system developed by MIT. It is the basis for other products, such as Enhanced X Windows which runs on the AIX operating system.





---

## Index

### Numerics

4MM device type 121  
8MM device type 124

### A

ACCESS parameter

See the DEFINE and UPDATE STGPOOL  
commands

See the DEFINE and UPDATE VOLUME commands

account record, setting 505

ACTIVATE POLICYSET command 46

activating policy sets 46

address

get IPX 639

administrative client

batch mode 13, 16

console mode 13, 14

continuation characters 19

interactive mode 13, 16

macros 21

mount mode 13, 15

options 18

privilege classes 29

using 13

administrative client options

-consolemode 18

-id 18

-itemcommit 18, 25

-mountmode 18

-noconfirm 18

-outfile 14, 15, 18

-password 18

-quiet 18

administrative command

components 5

entering 5

entry rules 6

formatting terminal output 8

help 280

issued by all administrators 40

issued with analyst privilege 39

issued with operator privilege 38

issued with policy privilege 33

issued with storage privilege 35

administrative command (*continued*)

issued with system privilege 29

issuing multiple 21

keyword parameters

optional 5

required 5

naming conventions 10

positional parameters

optional 5

required 5

privilege class for 29

processing

background 7

foreground 7

scheduling 171

using 45

using wildcard characters 8

administrative session, canceling 78

administrator

exporting 250

importing 283

locking 303

querying 317

registering 461

removing 470

renaming 474

updating 537

appending

command output to a file 21

application programming interface xvi

ARCHDELETE parameter

See the REGISTER and UPDATE NODE commands

archive file deletion via a client, allowing

See the REGISTER and UPDATE NODE commands

ASSIGN DEFMGMTCLASS command 48

assigning, default management class 48

association of client with schedule

defining 100

deleting 197

querying 322

AUDIT LIBRARY command 51

AUDIT LICENSES command 54

audit occupancy

querying 325

AUDIT VOLUME command 56

auditing

- auditing (*continued*)
  - library 51
  - licenses 54
  - storage pool volume 56
- authentication
  - setting 509
- authority
  - for command use 29
  - granting 273
  - revoking 498
  - unlocking 535
- AUTOMIGNONUSE parameter
  - See the DEFINE and UPDATE MGMTCLASS commands

## B

- BACKDELETE parameter
  - See the REGISTER and UPDATE NODE commands
- backup
  - database 61
  - database volume information 646
  - device information 66
  - primary storage pool 68
  - scheduling for client 164
  - trigger for database 112
  - volume history 72
- BACKUP DB command 61
- BACKUP DEVCONFIG command 66
- backup file deletion via a client, allowing
  - See the REGISTER and UPDATE NODE commands
- BACKUP STGPOOL command 68
- BACKUP VOLHISTORY command 72
- batch mode
  - restrictions 16
  - starting and ending 16
  - using 16
- BEGINDATE parameter
  - See the QUERY ACTLOG and QUERY EVENT commands
- BEGINTIME parameter
  - See the QUERY ACTLOG and QUERY EVENT commands
- buffer pool statistic, resetting 482
- BUFPOOLSIZE option 681

## C

- CACHE parameter
  - See the DEFINE and UPDATE STGPOOL commands

- CANCEL PROCESS command 74
- CANCEL REQUEST command 76
- CANCEL SESSION command 78
- canceling
  - administrative session 78
  - background process 8
  - client node session 78
  - mount requests 76
  - process 74
- CHECKIN LIBVOLUME command 80
- checking in
  - libvolume 80
- checking out
  - libvolume 84
- CHECKLABEL parameter
  - See the CHECKIN, CHECKOUT LIBVOLUME and AUDIT LIBRARY commands
- CHECKOUT LIBVOLUME command 84
- CLASSES parameter
  - See the GRANT and REVOKE AUTHORITY commands
- client node session
  - canceling 78
  - scheduling 164
- client-server options, setting 678
- COLLOCATE parameter
  - See the DEFINE and UPDATE STGPOOL commands
- collocation 186, 189
- command line interface
  - administrative client 13
  - server console 13
  - using 13
- commands
  - ACTIVATE POLICYSET 46
  - ASSIGN DEFMGMTCLASS 48
  - AUDIT LIBRARY 51
  - AUDIT LICENSES 54
  - AUDIT VOLUME 56
  - BACKUP DB 61
  - BACKUP DEVCONFIG 66
  - BACKUP STGPOOL 68
  - BACKUP VOLHISTORY 72
  - CANCEL PROCESS 74
  - CANCEL REQUEST 76
  - CANCEL SESSION 78
  - CHECKIN LIBVOLUME 80
  - CHECKOUT LIBVOLUME 84
  - COMMIT 87
  - COPY DOMAIN 89

commands (continued)

COPY MGMTCLASS 92  
COPY POLICYSET 94  
COPY SCHEDULE 96  
DEFINE ASSOCIATION 100  
DEFINE COPYGROUP 103  
DEFINE DBBACKUPTRIGGER 112  
DEFINE DBCOPY 116  
DEFINE DBVOLUME 118  
DEFINE DEVCLASS 120  
DEFINE DOMAIN 144  
DEFINE DRIVE 147  
DEFINE LIBRARY 150  
DEFINE LOGCOPY 153  
DEFINE LOGVOLUME 155  
DEFINE MGMTCLASS 157  
DEFINE POLICYSET 161  
DEFINE SCHEDULE 164  
DEFINE STGPOOL 180  
DEFINE VOLUME 193  
DELETE ASSOCIATION 197  
DELETE COPYGROUP 199  
DELETE DBBACKUPTRIGGER 202  
DELETE DBVOLUME 204  
DELETE DEVCLASS 206  
DELETE DOMAIN 208  
DELETE DRIVE 210  
DELETE EVENT 212  
DELETE FILESPACE 216  
DELETE LIBRARY 220  
DELETE LOGVOLUME 222  
DELETE MGMTCLASS 224  
DELETE POLICYSET 226  
DELETE SCHEDULE 229  
DELETE STGPOOL 232  
DELETE VOLHISTORY 234  
DELETE VOLUME 240  
DISABLE 243  
DISMOUNT VOLUME 245  
DSMFMT 634  
DSMLABEL 637  
DSMSERV AUDITDB 673  
DSMSERV DISPLAY DBBACKUPVOLUME 646  
DSMSERV DISPLAY DBVOLUMES 649  
DSMSERV DISPLAY LOGVOLUMES 651  
DSMSERV DUMPDB 662  
DSMSERV EXTEND LOG 642  
DSMSERV INSTALL 667  
DSMSERV LOADDDB 669  
DSMSERV RESTORE DB 653

commands (continued)

ENABLE 246  
EXPIRE INVENTORY 248  
EXPORT ADMIN 250  
EXPORT NODE 254  
EXPORT POLICY 260  
EXPORT SERVER 264  
EXTEND DB 269  
EXTEND LOG 271  
GETIPXAD 639  
GRANT AUTHORITY 273  
HALT 277  
HELP 280  
IMPORT ADMIN 283  
IMPORT NODE 287  
IMPORT POLICY 293  
IMPORT SERVER 297  
LOCK ADMIN 303  
LOCK NODE 304  
MACRO 306  
MOVE DATA 309  
QUERY ACTLOG 311  
QUERY ADMIN 317  
QUERY ASSOCIATION 322  
QUERY AUDITOCUPANCY 325  
QUERY CONTENT 328  
QUERY COPYGROUP 334  
QUERY DB 339  
QUERY DBBACKUPTRIGGER 345  
QUERY DBVOLUME 347  
QUERY DEVCLASS 350  
QUERY DOMAIN 354  
QUERY DRIVE 359  
QUERY EVENT 362  
QUERY FILESPACE 377  
QUERY LIBRARY 381  
QUERY LIBVOLUME 384  
QUERY LICENSE 386  
QUERY LOG 389  
QUERY LOGVOLUME 394  
QUERY MGMTCLASS 397  
QUERY MOUNT 401  
QUERY NODE 403  
QUERY OCCUPANCY 408  
QUERY OPTION 412  
QUERY POLICYSET 415  
QUERY PROCESS 419  
QUERY REQUEST 421  
QUERY SCHEDULE 422  
QUERY SESSION 427

commands (*continued*)

QUERY STATUS 431  
QUERY STGPOOL 436  
QUERY VOLHISTORY 443  
QUERY VOLUME 450  
QUIT 456  
REDUCE DB 457  
REDUCE LOG 459  
REGISTER ADMIN 461  
REGISTER LICENSE 463  
REGISTER NODE 466  
REMOVE ADMIN 470  
REMOVE NODE 472  
RENAME ADMIN 474  
RENAME FILESPACE 476  
RENAME NODE 479  
RESET BUFPOOL 482  
RESET DBMAXUTILIZATION 484  
RESET LOGCONSUMPTION 485  
RESET LOGMAXUTILIZATION 487  
RESTORE STGPOOL 488  
RESTORE VOLUME 493  
REVOKE AUTHORITY 498  
ROLLBACK 503  
SET ACCOUNTING 505  
SET ACTLOGRETENTION 507  
SET AUTHENTICATION 509  
SET EVENTRETENTION 510  
SET LICENSEAUDITPERIOD 512  
SET LOGMODE 514  
SET MAXCMDRETRIES 517  
SET MAXSCHEDSESSIONS 519  
SET PASSEXP 521  
SET QUERYSCHEDPERIOD 524  
SET RANDOMIZE 526  
SET REGISTRATION 528  
SET RETRYPERIOD 530  
SET SCHEDMODES 532  
SET SERVERNAME 534  
UNLOCK ADMIN 535  
UNLOCK NODE 536  
UPDATE ADMIN 537  
UPDATE COPYGROUP 539  
UPDATE DBBACKUPTRIGGER 547  
UPDATE DEVCLASS 550  
UPDATE DOMAIN 573  
UPDATE DRIVE 575  
UPDATE LIBRARY 577  
UPDATE LIBVOLUME 579  
UPDATE MGMTCLASS 581

commands (*continued*)

UPDATE NODE 585  
UPDATE POLICYSET 588  
UPDATE SCHEDULE 590  
UPDATE STGPOOL 605  
UPDATE VOLHISTORY 617  
UPDATE VOLUME 619  
UPGRADEDDB 640  
VALIDATE POLICYSET 625  
VARY 629  
commands in a macro  
  committing 87  
  rolling back 503  
  when to use 22  
COMMIT command 87  
committing  
  commands in a macro 87  
COMMOPENTIMEOUT option 681  
COMMTIMEOUT option 682  
console mode  
  ending 15  
  restrictions 14  
  starting 14  
  using 14  
continuation characters  
  for a list of values 19  
  for a quoted list of values 19  
  using 19  
COPIED parameter  
  See the QUERY CONTENT command  
COPY DOMAIN command 89  
copy group  
  defining archive 103  
  defining backup 103  
  deleting 199  
  description of 103  
  querying 334  
  restriction 103  
  updating archive 539  
  updating backup 539  
COPY MGMTCLASS command 92  
COPY POLICYSET command 94  
COPY SCHEDULE command 96  
copying  
  management class 92  
  policy domain 89  
  policy set 94  
  schedule 96  
COUNT parameter  
  See the QUERY CONTENT command

## D

DAMAGED parameter

See the QUERY CONTENT command

data

moving 309

database

auditing 673

backup 61

backup trigger 112

backup volume information 646

dumping 662

extending 269

installing 667

querying 339

reducing 457

reloading 669

resetting maximum utilization statistic 484

restoring 653

setting options 679

verifying 673

volume information 347, 649

database maximum utilization statistic

resetting 484

database volume

defining 118

deleting 204

displaying information 649

querying 347

date format 679

DATEFORMAT option 682

DAYOFWEEK parameter

See the DEFINE and UPDATE SCHEDULE commands

default management class

assigning 48

description of 157

DEFINE ASSOCIATION command 100

DEFINE COPYGROUP command 103

DEFINE DBBACKUPTRIGGER command 112

DEFINE DBCOPY command 116

DEFINE DBVOLUME command 118

DEFINE DEVCLASS command 120

DEFINE DOMAIN command 144

DEFINE DRIVE command 147

DEFINE LIBRARY command 150

DEFINE LOGCOPY command 153

DEFINE LOGVOLUME command 155

DEFINE MGMTCLASS command 157

DEFINE POLICYSET command 161

DEFINE SCHEDULE command 164

DEFINE STGPOOL command 180

DEFINE VOLUME command 193

defining

association 100

copy group 103

database backup trigger 112

database volume 118

device class 120

domain 144

drive 147

library 150

management class 157

mirrored database volume 116

mirrored log volume 153

policy set 161

recovery log volume 155

schedule 164

storage pool 180

storage pool volume 193

DELETE ASSOCIATION command 197

DELETE COPYGROUP command 199

DELETE DBBACKUPTRIGGER command 202

DELETE DBVOLUME command 204

DELETE DEVCLASS command 206

DELETE DOMAIN command 208

DELETE DRIVE command 210

DELETE EVENT command 212

DELETE FILESPACE command 216

DELETE LIBRARY command 220

DELETE LOGVOLUME command 222

DELETE MGMTCLASS command 224

DELETE POLICYSET command 226

DELETE SCHEDULE command 229

DELETE STGPOOL command 232

DELETE VOLHISTORY command 234

DELETE VOLUME command 240

deleting

archive file deletion via a client, allowing

See the REGISTER and UPDATE NODE commands

backup file deletion via a client, allowing

See the REGISTER and UPDATE NODE commands

copy group 199

database backup trigger 202

database volume 204

device class 206

domain 208

drive 210

- deleting (*continued*)
  - event record 212
  - file space 216
  - library 220
  - management class 224
  - policy set 226
  - recovery log volume 222
  - schedule 229
  - storage pool 232
  - storage pool volume 240
  - volume history 234
- device class
  - defining 120
  - deleting 206
  - querying 350
  - updating 550
- device configuration file 66
- DISABLE command 243
- disabling the server 243
- DISCARDATA parameter
  - See the DELETE VOLUME command
- DISMOUNT VOLUME command 245
- dismounting volumes 245
- DLT device type 127
- documentation, user xv
- domain
  - copy 89
  - defining 144
  - deleting 208
  - querying 354
  - updating 573
- drive
  - defining 147, 359
  - deleting 210
  - updating 575
- DSMFMT command 634
- DSMLABEL command 637
- DSMSERV AUDITDB command 673
- DSMSERV DISPLAY DBBACKUPVOLUME command 646
- DSMSERV DISPLAY DBVOLUMES command 649
- DSMSERV DISPLAY LOGVOLUMES command 651
- DSMSERV DUMPDB command 662
- DSMSERV EXTEND LOG command 642
- DSMSERV INSTALL command 667
- DSMSERV LOADDDB command 669
- DSMSERV RESTORE DB command 653
- dumping
  - database 662

- DURATION parameter
  - See the DEFINE and UPDATE SCHEDULE commands

## E

- ENABLE command 246
- enabling the server 246
- ENDDATE parameter
  - See the QUERY ACTLOG and QUERY EVENT commands
- ending
  - batch mode 16
  - console mode 15
  - interactive mode 18
  - mount mode 16
- ENDTIME parameter
  - See the QUERY ACTLOG and QUERY EVENT commands
- entering administrative commands 5
- ESTCAPACITY parameter
  - See the DEFINE and UPDATE DEVCLASS commands
- event record, deleting 212
- event, querying 362
- EXCEPTIONSONLY parameter
  - See the QUERY EVENT command
- EXPINTERVAL option 683
- EXPIRATION parameter
  - See the DEFINE and UPDATE SCHEDULE commands
- EXPIRE INVENTORY command 248
- expiring inventory 248
- export
  - administrator 250
  - node 254
  - policy information 260
  - server 264
- EXPORT ADMIN command 250
- EXPORT NODE command 254
- EXPORT POLICY command 260
- EXPORT SERVER command 264
- EXPQUIET option 684
- EXTEND DB command 269
- EXTEND LOG command 271
- extending
  - database 269
  - recovery log 271

## F

- FILE device type 130
- FILEDATA parameter
  - See the EXPORT and IMPORT NODE commands
- FILENAMES parameter
  - See the BACKUP VOLHISTORY and BACKUP DEVCONFIG commands
- files
  - collocating by client 180
  - formatting for server use 634, 640
  - moving 309
  - removing expired files 248
- filesystem
  - deleting 216
  - querying 377
  - renaming 476
- FILESIZE parameter
  - See the EXPORT and IMPORT NODE commands
- FORCE parameter
  - See the CHECKOUT LIBVOLUME command
- FORMAT parameter
  - See the DEFINE and UPDATE DEVCLASS commands
- formatting files for the server 634, 640
- FREQUENCY parameter
  - See the DEFINE and UPDATE COPYGROUP commands

## G

- GETIPXAD command 639
- GRANT AUTHORITY command 273
- granting
  - authority 273
  - types of privilege classes 273

## H

- HALT command 277
- halting the server 277
- help
  - administrative commands 280
- HELP command 280
- hierarchical storage management
  - DEFINE MGMTCLASS 157
  - description of 157
  - UPDATE MGMTCLASS 581
- hierarchy of storage pools
  - considerations for backup 68

- hierarchy of storage pools (*continued*)
  - defining 180
- HIGHMIG parameter
  - See the DEFINE and UPDATE STGPOOL commands
- HSM
  - See hierarchical storage management

## I

- IDLETIMEOUT option 685
- IMPORT ADMIN command 283
- IMPORT NODE command 287
- IMPORT POLICY command 293
- IMPORT SERVER command 297
- importing
  - administrator 283
  - node 287
  - policy information 293
  - server 297
- INCRDEVCLASS parameter
  - See the DEFINE AND UPDATE DBBACKUPTRIGGER commands
- incremental backup of database 61
- installing
  - database 667
  - recovery log 667
- interactive mode
  - continuation characters 19
  - ending 18
  - quitting 456
  - restrictions 16
  - starting 17
  - using 16
- inventory
  - expiring 248
- IPX
  - get address 639
- IPXBUFFERSIZE option 685
- IPXSocket option 685

## L

- labels
  - storage volumes 637
- LANAdapter option 686
- language format 679
- LANGUAGE option 686
- library
  - auditing 51

- library (*continued*)
  - defining 150
  - deleting 220
  - querying 381
  - updating 577
- libvolume
  - checking in 80
  - checking out 84
  - querying 384
  - updating 579
- license
  - auditing 54
  - querying 386
  - registering 463
  - setting audit period 512
- license audit period
- LOCK ADMIN command 303
- LOCK NODE command 304
- locking
  - administrator 303
  - node 304
- log mode, setting 514
- LOGFULLPCT parameter
  - See the DEFINE AND UPDATE DBBACKUPTRIGGER commands
- LOGPOOLSIZE option 687
- LOWMIG parameter
  - See the DEFINE and UPDATE STGPOOL commands
- LUNAME option 687

## M

- MACRO command 306
- macros
  - commit individual commands 25
  - continuation characters 19
  - controlling command processing 25
  - running 24
  - substitution variables 24
  - testing 25
  - using 21, 306
  - writing commands 23
  - writing comments 23
- management class
  - copying 92
  - defining the default 48
  - deleting 224
  - querying 397
  - updating 581

- maximum retries
- maximum retries, setting 517
- MAXPROCESS parameter
  - See the BACKUP and RESTORE STGPOOL commands
- MAXSCRATCH parameter
  - See the DEFINE and UPDATE STGPOOL commands
- MAXSESSIONS option 687
- MAXSIZE parameter
  - See the DEFINE and UPDATE STGPOOL commands
- MIGDESTINATION parameter
  - See the DEFINE and UPDATE MGMTCLASS commands
- MIGPROCESS parameter
  - See the DEFINE and UPDATE STGPOOL commands
- migration
  - files from client node 157
  - storage pool 180
- MIGREQUIRESBKUP parameter
  - See the DEFINE and UPDATE MGMTCLASS commands
- mirrored database volume
  - defining 116
- mirrored log volume
  - defining 153
- MIRRORRead option 689
- MIRRORWrite option 690
- MODE parameter
  - See the DEFINE and UPDATE COPYGROUP commands
- monitoring
  - ADSM activities 14, 15
- mount mode
  - ending 16
  - using 15
- mount request
  - canceling 76
  - querying 421
- mounted sequential access volumes 401
  - dismounting 245
  - querying 401
- MOUNTLIMIT parameter
  - See the DEFINE and UPDATE DEVCLASS commands
- MOUNTRETENTION parameter
  - See the DEFINE and UPDATE DEVCLASS commands



MOUNTWAIT parameter  
See the CHECKIN LIBVOLUME command  
See the DEFINE and UPDATE DEVCLASS commands  
MOVE DATA command 309  
MOVEBatchsize 690  
MOVESizethresh 691  
moving  
files 309  
MSGNO parameter  
See the QUERY ACTLOG command

## N

NAMEdpipeoption option 691  
naming conventions  
case-sensitivity 10  
introduction 10  
naming volumes 10  
restrictions 11  
NETBIOSBuffersize option 692  
NETBiosname option 692  
NETBIOSSession option 693  
NEXTSTGPOOL parameter  
See the DEFINE and UPDATE STGPOOL commands  
NOAUDITSTORAGE option 693  
node  
associating with a schedule 100  
exporting 254  
importing 287  
locking 304  
querying 403  
registering 466  
removing 472  
renaming 479  
unlocking 536  
updating 585  
normal recovery log mode 61, 514  
number format 679  
NUMBERFORMAT option 693  
NUMINCREMENTAL parameter  
See the DEFINE AND UPDATE DBBACKUPTRIGGER commands

## O

OBJECTS parameter  
See the DEFINE and UPDATE SCHEDULE commands

occupancy  
querying 408  
OPTICAL device type 136, 565  
OPTIONS parameter  
See the DEFINE and UPDATE SCHEDULE commands  
options, server  
importance for recovery 61  
querying 412  
OWNER parameter  
See the DELETE FILESPACE command

## P

password  
setting expiration interval 521  
UPDATE ADMIN command 537  
pattern-matching expression  
creating 8  
PERUNITS parameter  
See the DEFINE and UPDATE SCHEDULE commands  
policy privilege class  
administrative commands 33  
restricted 33  
unrestricted 33  
policy set  
activating 46  
copying 94  
defining 161  
deleting 226  
querying 415  
updating 588  
validating 625  
polling  
information about 524  
setting frequency 524  
PREFIX parameter  
See the DEFINE and UPDATE DEVCLASS commands  
PREVIEW parameter  
See the EXPORT and IMPORT commands  
See the STRRSTADSM command  
See the UPDATE VOLUME command  
printing  
redirected output 21  
privilege class  
administrative commands 35, 38  
administrator issued 41  
analyst 29, 39

privilege class (*continued*)

- none 40
- operator 29, 39
- policy 29, 33
- restricted 35
- storage 29, 36
- system 29
- unrestricted 35

process

- canceling 74
- querying 419

program directory xv

publications xv

## Q

QIC device type 138

QUERY ACTLOG command 311

QUERY ADMIN command 317

QUERY ASSOCIATION command 322

QUERY AUDITOCCUPANCY command 325

QUERY CONTENT command 328

QUERY COPYGROUP command 334

QUERY DB command 339

QUERY DBBACKUPTRIGGER command 345

QUERY DBVOLUME command 347

QUERY DEVCLASS command 350

QUERY DOMAIN command 354

QUERY DRIVE command 359

QUERY EVENT command 362

QUERY FILESPACE command 377

QUERY LIBRARY command 381

QUERY LIBVOLUME command 384

QUERY LICENSE command 386

QUERY LOG command 389

QUERY LOGVOLUME command 394

QUERY MGMTCLASS command 397

QUERY MOUNT command 401

QUERY NODE command 403

QUERY OCCUPANCY command 408

QUERY OPTION command 412

QUERY OPTIONS command 681

QUERY POLICYSET command 415

QUERY PROCESS command 419

QUERY REQUEST command 421

QUERY SCHEDULE command 422

QUERY SESSION command 427

QUERY STATUS command 431

QUERY STGPOOL command 436

QUERY VOLHISTORY command 443

QUERY VOLUME command 450

querying

- activity log 311

- administrator 317

- audit occupancy 325

- background process 8

- clients with schedules 322

- completed events 362

- contents of a volume 328

- copy group 334

- database 339

- database backup trigger 345

- database volume 347

- device class 350

- domain 354

- drive 359

- file space 377

- library 381

- libvolume 384

- license 386

- management class 397

- mount requests 421

- node 403

- occupancy 408

- option 412

- policy set 415

- process 419

- recovery log 389

- recovery log volume 394

- scheduled events 362

- schedules 422

- session 427

- status 431

- storage pool 436

- storage pool volume 450

- volume history file 443

QUIT command 456

quitting

- interactive mode 456

## R

RECLAIM parameter

- See the DEFINE and UPDATE STGPOOL commands

reclamation of volumes 187, 190

recovery log

- backup trigger 112

- consumption statistic, resetting 485

- recovery log (*continued*)
  - extending 271
  - installing 667
  - maximum utilization statistic, resetting 487
  - reducing 459
  - setting mode 514
  - setting options 679
  - volume information 651
- recovery log volume
  - defining 155
  - deleting 222
  - querying 394
- redirection
  - appending output 21
  - command output 19
  - writing output 21
- redirection characters
  - using 19
- REDUCE DB command 457
- REDUCE LOG command 459
- reducing
  - database 457
  - recovery log 459
- REGISTER ADMIN command 461
- REGISTER LICENSE command 463
- REGISTER NODE command 466
- registering
  - administrator 461
  - license 463
  - node 466
- registration
  - closed 528
  - description of 528
  - open 528
  - setting 528
- reloading
  - database 669
- REMOVABLEFILE device type 141, 570
- REMOVE ADMIN command 470
- REMOVE NODE command 472
- REMOVE parameter
  - See the CHECKOUT LIBVOLUME command
- removing
  - administrator 470
  - client association 197
  - node 472
- RENAME ADMIN command 474
- RENAME FILESPACE command 476
- RENAME NODE command 479

- renaming
  - administrator 474
  - file space 476
  - node 479
- REPLACEDEFS parameter
  - See the IMPORT ADMIN and IMPORT POLICY commands
- REPLY command 481
- request
  - See mount request
- RESET BUFPOOL command 482
- RESET DBMAXUTILIZATION command 484
- RESET LOGCONSUMPTION command 485
- RESET LOGMAXUTILIZATION command 487
- resetting
  - buffer pool statistic 482
  - maximum utilization statistic (recovery log) 487
  - recovery log consumption statistic 485
- RESTORE STGPOOL command 488
- RESTORE VOLUME command 493
- retention period
  - description of 510
  - setting 510
- RETONLY parameter
  - See the DEFINE and UPDATE COPYGROUP commands
- retry period
  - description of 530
  - setting 530
- RETVER parameter
  - See the DEFINE and UPDATE COPYGROUP commands
- REUSEDELAY parameter
  - See the DEFINE and UPDATE STGPOOL commands
- REVOKE AUTHORITY command 498
- revoking
  - authority 498
- roll-forward recovery log mode 61, 514
- ROLLBACK command 503
- rolling back
  - commands in a macro 503

## S

- schedule
  - administrative command 171
  - associating with a client node 100
  - client 165
  - copying 96

- schedule (*continued*)
  - defining 157, 164
  - deleting 229
  - description of 171
  - querying 422
  - querying results of (events) 362
  - restrictions of 164
  - startup window 167
  - types of 164
  - updating 590
- scheduling mode
  - information about 532
  - setting 532
- scratch volumes in storage pool 187, 190
- SERIALIZATION parameter
  - See the DEFINE and UPDATE COPYGROUP commands
- server
  - disabling 243
  - enabling 246
  - exporting 264
  - importing 297
- server console
  - restrictions 13
  - using 13
- server name
  - setting 534
- server options
  - modifying the file 677
  - querying 681
  - setting the date format 679
  - setting the language format 679
  - setting the number format 679
  - setting the time format 679
  - tailoring 677
- session
  - maximum number scheduled 519
  - querying 427
- SET ACCOUNTING command 505
- SET ACTLOGRETENTION command 507
- SET AUTHENTICATION command 509
- SET EVENTRETENTION 510
- SET LICENSEAUDITPERIOD command 512
- SET LOGMODE command 514
- SET MAXCMDRETRIES command 517
- SET MAXSCHEDSESSIONS command 519
- SET PASSEXP command 521
- SET QUERYSCHEDPERIOD 524
- SET RANDOMIZE command 526
- SET REGISTRATION command 528
- SET RETRYPERIOD command 530
- SET SCHEDMODES 532
- SET SERVERNAME 534
- setting
  - accounting record 505
  - authentication 509
  - frequency for client-polling 524
  - license audit period 512
  - maximum retries of a command 517
  - maximum scheduled sessions 519
  - password 521
  - randomization of start times 526
  - recovery log mode 514
  - registration 528
  - retention period 507
  - retention period for event records 510
  - retry period 530
  - scheduling mode 532
  - server name 534
- setting communication options 677
- SHMQUEUEUENAME option 694
- SKIPPARTIAL parameter
  - See the AUDIT VOLUME command
- SNABuffersize 694
- SPACEMGTECHNIQUE parameter
  - See the DEFINE and UPDATE MGMTCLASS commands
- start time
  - information about 164
  - setting 526
- STARTDATE parameter
  - See the DEFINE and UPDATE SCHEDULE commands
- starting
  - batch mode 16
  - console mode 14
  - interactive mode 17
  - mount mode 15
- STARTTIME parameter
  - See the DEFINE and UPDATE SCHEDULE commands
- status
  - querying 431
- STAtusmsgcnt 695
- STGPOOL parameter
  - See the MOVE DATA command
- STGPOOLS parameter
  - See the GRANT and REVOKE AUTHORITY commands

- storage management services xv
- storage pool
  - defining 180
  - deleting 232
  - querying 436
  - restoring 488
  - updating 605
- storage pool volume
  - auditing 56
  - defining 193
  - deleting 240
  - querying 328, 450
  - scratch volumes in storage pool 187, 190
  - updating 619
  - varying 629
- storage volumes
  - labeling 637
  - naming 10
- SWAP parameter
  - See the CHECKIN LIBVOLUME command
- syntax diagram
  - abbreviations 1
  - default value 2
  - fragments 3
  - optional choice 2
  - repeatable choice 2
  - repeating values 2
  - required choices 2
  - symbols 1
  - using 1
  - variables 1
- system privilege class
  - administrative commands 29

## T

- TCPNODELAY option 695
- TCPPORT option 696
- TCPWindowSize option 696
- terminal output, formatting 8
- time format 679
- TIMEFORMAT option 696
- trademarks xii
- trigger, for database backup 112, 345

## U

- UNLOCK ADMIN command 535
- UNLOCK NODE command 536

- unlocking
  - administrator 535
  - node 536
- UPDATE ADMIN command 537
- UPDATE COPYGROUP command 539
- UPDATE DBBACKUPTRIGGER command 547
- UPDATE DEVCLASS command 550
- UPDATE DOMAIN command 573
- UPDATE DRIVE command 575
- UPDATE LIBRARY command 577
- UPDATE LIBVOLUME command 579
- UPDATE MGMTCLASS command 581
- UPDATE NODE command 585
- UPDATE POLICYSET command 588
- UPDATE SCHEDULE command 590
- UPDATE STGPOOL command 605
- UPDATE VOLHISTORY 617
- UPDATE VOLUME command 619
- updating
  - administrator 537
  - copy group 539
  - database backup trigger 547
  - device class 550
  - domain 573
  - drive 575
  - library 577
  - libvolume 579
  - management class 581
  - node 585
  - policy set 588
  - schedule 590
  - storage pool 605
  - storage pool volume 619
  - volume history 617
- UPGRADEDDB command 640
- user documentation xv
- using
  - administrative client options 18
  - administrative privilege classes 29
  - command line interface 13
  - commands by administrative privilege 29
  - continuation characters 19
  - macros 306
  - redirection 19
  - redirection characters 19
  - syntax diagram 1

## V

- VALIDATE POLICYSET command 625
- validating
  - policy set 625
- VARY command 629
- varying
  - volume 629
- VERDELETED parameter
  - See the DEFINE and UPDATE COPYGROUP commands
- VEREXISTS parameter
  - See the DEFINE and UPDATE COPYGROUP commands
- verifying the database 673
- volume history
  - backing up 72
  - deleting 234
  - querying 443
- VOLUMEHISTORY option 698
- volumes, storage
  - defining 193
  - labeling 637
  - naming 10
  - reclaiming 187, 190
  - restoring 493
  - scratch, maximum in storage pool 187, 190

## W

- WHEREACCESS parameter
  - See the UPDATE VOLUME command
- WHERESTATUS parameter
  - See the UPDATE VOLUME command
- WHERESTGPPOOL parameter
  - See the UPDATE VOLUME command
- wildcard
  - match-any examples 9
  - match-exactly-one examples 10
  - using 8
- writing
  - command output to a file 21



---

## Communicating Your Comments to IBM

ADSTAR Distributed Storage Manager  
for OS/2  
Administrator's Reference  
Version 2  
Publication No. GC35-0233-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States and Canada: 520 799-2906
  - Other countries: (1) 520 799-2906

The contact department is 61C/031.

- If you prefer to send comments by electronic mail, use one of the following addresses:
  - Internet: [starpubs@vnet.ibm.com](mailto:starpubs@vnet.ibm.com) (or `starpubs` at `vnet.ibm.com`)
  - IBMLink from U.S.A.: STARPUBS at SJEVM5
  - IBMLink from Canada: STARPUBS at TORIBM
  - IBM Mail Exchange: USIB3VVD at IBMMAIL

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.



---

## Readers' Comments — We'd Like to Hear from You

**ADSTAR Distributed Storage Manager  
for OS/2  
Administrator's Reference  
Version 2**

**Publication No. GC35-0233-00**

**Overall, how satisfied are you with the information in this book?**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**How satisfied are you that the information in this book is:**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you?  Yes  No

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

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_  
Phone No.

Readers' Comments — We'd Like to Hear from You  
GC35-0233-00



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



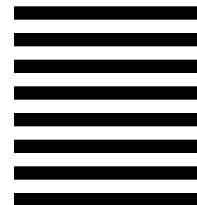
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Information Development  
Department 61C  
9000 South Rita Road  
TUCSON AZ 85775-4401



Fold and Tape

Please do not staple

Fold and Tape

GC35-0233-00

Cut or Fold  
Along Line





Program Number: 84H3121



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

GC35-0233-00



*Spine information:*



ADSTAR Distributed Storage Manager  
for OS/2

Administrator's Reference

*Version 2*