

Version 6.1



## Administrator's Reference





## Administrator's Reference

**Note**

Before using this information and the product it supports, read the information in “Notices” on page 1281.

This edition applies to Version 6.1 of IBM Tivoli Storage Manager and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

© **Copyright International Business Machines Corporation 1993, 2009.**

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

---

# Contents

## Preface . . . . . xi

|  |      |
|--|------|
| Who should read this publication . . . . .     | xi   |
| Publications . . . . .                         | xi   |
| Tivoli Storage Manager publications . . . . .  | xi   |
| Related hardware publications . . . . .        | xiii |
| Support information . . . . .                  | xiii |
| Getting technical training . . . . .           | xiv  |
| Searching knowledge bases . . . . .            | xiv  |
| Contacting IBM Software Support . . . . .      | xv   |
| Conventions used in this publication . . . . . | xvii |

## New for IBM Tivoli Storage Manager

### Version 6.1 . . . . . xix

|   |        |
|---|--------|
| New for the server in Version 6.1.2 . . . . .                     | xix    |
| Enabled functions . . . . .                                       | xix    |
| Licensing changes . . . . .                                       | xix    |
| TSMDLST utility for AIX . . . . .                                 | xx     |
| Device support for AIX . . . . .                                  | xx     |
| <b>PREVIEW</b> parameter for DSMSEVER INSERTDB . . . . .          | xx     |
| New server option SANDISCOVERYTIMEOUT . . . . .                   | xx     |
| New for the server in Version 6.1.0 . . . . .                     | xx     |
| Disabled functions in 6.1.0 and 6.1.1 . . . . .                   | xxi    |
| Changes to the Version 6.1 Administration Center . . . . .        | xxi    |
| Data deduplication . . . . .                                      | xxiii  |
| Storage devices . . . . .   | xxiv   |
| Disaster recovery manager support for active-data pools . . . . . | xxv    |
| EXPIRE INVENTORY command enhancements . . . . .                   | xxvi   |
| No-query restore changes . . . . .                                | xxvii  |
| Server database . . . . .   | xxvii  |
| Support for NetApp SnapMirror to Tape feature . . . . .           | xxvii  |
| Reporting and monitoring feature . . . . .                        | xxvii  |
| ODBC driver support . . . . .                                     | xxviii |
| Backup sets and client node enhancements . . . . .                | xxviii |

## Command and option changes . . . . . xxix

|   |        |
|---|--------|
| New server commands, utilities, and options . . . . .     | xxix   |
| Updated server commands, utilities, and options . . . . . | xxxi   |
| Deleted server commands, utilities, and options . . . . . | xxxiv  |
| Changes to the SELECT command . . . . .                   | xxxvii |

## Chapter 1. Administering the server from the command line . . . . . 1

|  |   |
|--|---|
| Issuing commands from the administrative client . . . . .                  | 1 |
| Starting and stopping the administrative client . . . . .                  | 2 |
| Monitoring server activities from the administrative client . . . . .      | 2 |
| Monitoring removable-media mounts from the administrative client . . . . . | 2 |
| Processing commands one at a time from the administrative client . . . . . | 3 |

|  |    |
|--|----|
| Processing a series of commands from the administrative client . . . . . | 3  |
| Formatting output from commands . . . . .                                | 4  |
| Saving command output to a specified location . . . . .                  | 4  |
| Administrative client options . . . . .                                  | 5  |
| Issuing commands from the Administration Center . . . . .                | 7  |
| Issuing commands from the server console . . . . .                       | 7  |
| Entering administrative commands . . . . .                               | 8  |
| Reading syntax diagrams . . . . .  | 8  |
| Entering long commands . . . . .   | 11 |
| Naming Tivoli Storage Manager objects . . . . .                          | 12 |
| Using wildcard characters to specify object names . . . . .              | 13 |
| Specifying descriptions in keyword parameters . . . . .                  | 14 |
| Controlling command processing . . . . .                                 | 14 |
| Server command processing . . . . .                                      | 14 |
| Cancelling commands . . . . .  | 15 |
| Performing tasks concurrently on multiple servers . . . . .              | 16 |
| Routing commands to a single server . . . . .                            | 16 |
| Routing commands to multiple servers . . . . .                           | 16 |
| Routing commands to a server group . . . . .                             | 17 |
| Routing commands to server groups . . . . .                              | 17 |
| Routing commands to two servers and a server group . . . . .             | 17 |
| Routing commands inside scripts . . . . .                                | 17 |
| Privilege classes for commands . . . . .                                 | 18 |
| Commands requiring system privilege . . . . .                            | 18 |
| Commands requiring policy privilege . . . . .                            | 21 |
| Commands requiring storage privilege . . . . .                           | 22 |
| Commands requiring operator privilege . . . . .                          | 23 |
| Commands any administrator can issue . . . . .                           | 24 |

## Chapter 2. Administrative commands 27

|   |    |
|---|----|
| ACCEPT DATE (Accepts the current system date) . . . . .                               | 28 |
| ACTIVATE POLICYSET (Activate a new policy set) . . . . .                              | 29 |
| ASSIGN DEFMGMTCLASS (Assign a default management class) . . . . .                     | 31 |
| AUDIT commands . . . . .  | 33 |
| AUDIT LIBRARY (Audit volume inventories in an automated library) . . . . .            | 34 |
| AUDIT LICENSES (Audit server storage usage) . . . . .                                 | 36 |
| AUDIT VOLUME (Verify database information for a storage pool volume) . . . . .        | 37 |
| BACKUP commands . . . . .   | 43 |
| BACKUP DB (Back up the database) . . . . .  | 44 |
| BACKUP DEVCONFIG (Create backup copies of device configuration information) . . . . . | 47 |
| BACKUP NODE (Back up a NAS node) . . . . .  | 49 |
| BACKUP STGPOOL (Back up primary storage pool to copy storage pool) . . . . .          | 54 |
| BACKUP VOLHISTORY (Save sequential volume history information) . . . . .              | 58 |
| BEGIN EVENTLOGGING (Begin logging events) . . . . .                                   | 60 |
| CANCEL commands . . . . .   | 62 |

|  |     |  |     |
|--|-----|--|-----|
| CANCEL EXPIRATION (Cancel an expiration process) . . . . .   | 63  | DEFINE MACHNODEASSOCIATION (Associate a node with a machine) . . . . .                 | 233 |
| CANCEL EXPORT (Delete a suspended export operation) . . . . .  | 64  | DEFINE MGMTCLASS (Define a management class) . . . . .                                 | 235 |
| CANCEL PROCESS (Cancel an administrative process) . . . . .  | 65  | DEFINE NODEGROUP (Define a node group) . . . . .                                       | 238 |
| CANCEL REQUEST (Cancel one or more mount requests) . . . . .   | 67  | DEFINE NODEGROUPMEMBER (Define node group member). . . . .                             | 239 |
| CANCEL RESTORE (Cancel a restartable restore session). . . . .   | 68  | DEFINE PATH (Define a path) . . . . .  | 240 |
| CANCEL SESSION (Cancel one or more client sessions) . . . . .  | 69  | DEFINE POLICYSET (Define a policy set). . . . .  | 246 |
| CHECKIN LIBVOLUME (Check a storage volume into a library). . . . .                                       | 70  | DEFINE PROFASSOCIATION (Define a profile association) . . . . .                        | 248 |
| CHECKOUT LIBVOLUME (Check a storage volume out of a library) . . . . .                                   | 77  | DEFINE PROFILE (Define a profile). . . . .   | 254 |
| CLEAN DRIVE (Clean a drive). . . . .   | 83  | DEFINE RECMEDMACHASSOCIATION (Associate recovery media with a machine) . . . . .       | 256 |
| COMMIT (Control committing of commands in a macro) . . . . .   | 84  | DEFINE RECOVERYMEDIA (Define recovery media) . . . . .                                 | 258 |
| COPY commands . . . . .  | 85  | DEFINE SCHEDULE (Define a client or an administrative command schedule) . . . . .      | 260 |
| COPY ACTIVATEDATA (Copy active backup data from a primary storage pool to an active-data pool) . . . . . | 86  | DEFINE SCRIPT (Define a Tivoli Storage Manager script) . . . . .                       | 281 |
| COPY CLOPTSET (Copy a client option set) . . . . .   | 90  | DEFINE SERVER (Define a server for server-to-server communications). . . . .           | 283 |
| COPY DOMAIN (Copy a policy domain) . . . . .   | 91  | DEFINE SERVERGROUP (Define a server group). . . . .                                    | 288 |
| COPY MGMTCLASS (Copy a management class) . . . . .   | 93  | DEFINE SPACETRIGGER (Define the space trigger) . . . . .                               | 289 |
| COPY POLICYSET (Copy a policy set) . . . . .   | 95  | DEFINE STGPOOL (Define a storage pool) . . . . .                                       | 292 |
| COPY PROFILE (Copy a profile) . . . . .  | 97  | DEFINE SUBSCRIPTION (Define a profile subscription) . . . . .                          | 333 |
| COPY SCHEDULE (Copy a client or an administrative command schedule) . . . . .                            | 99  | DEFINE VIRTUALFSMAPPING (Define a virtual file space mapping) . . . . .                | 335 |
| COPY SCRIPT (Copy a Tivoli Storage Manager script) . . . . .   | 103 | DEFINE VOLUME (Define a volume in a storage pool) . . . . .                            | 337 |
| COPY SERVERGROUP (Copy a server group) . . . . .   | 104 | DELETE commands . . . . .  | 342 |
| DEFINE commands . . . . .  | 105 | DELETE ASSOCIATION (Delete the node association to a schedule) . . . . .               | 344 |
| DEFINE ASSOCIATION (Associate client nodes with a schedule) . . . . .                                    | 106 | DELETE BACKUPSET (Delete a backup set) . . . . .                                       | 346 |
| DEFINE BACKUPSET (Define a backup set) . . . . .   | 108 | DELETE CLIENTOPT (Delete an option in an option set) . . . . .                         | 351 |
| DEFINE CLIENTACTION (Define a one-time client action). . . . .   | 113 | DELETE CLOPTSET (Delete a client option set) . . . . .                                 | 352 |
| DEFINE CLIENTOPT (Define an option to an option set) . . . . .   | 118 | DELETE COLLOCGROUP (Delete a collocation group). . . . .                               | 353 |
| DEFINE CLOPTSET (Define a client option set name) . . . . .  | 121 | DELETE COLLOCMEMBER (Delete collocation group member). . . . .                         | 354 |
| DEFINE COLLOCGROUP (Define a collocation group). . . . .   | 122 | DELETE COPYGROUP (Delete a backup or archive copy group) . . . . .                     | 356 |
| DEFINE COLLOCMEMBER (Define collocation group member). . . . .   | 124 | DELETE DATAMOVER (Delete a data mover) . . . . .                                       | 358 |
| DEFINE COPYGROUP (Define a copy group) . . . . .   | 126 | DELETE DEVCLASS (Delete a device class) . . . . .                                      | 359 |
| DEFINE DATAMOVER (Define a data mover) . . . . .   | 136 | DELETE DOMAIN (Delete a policy domain) . . . . .                                       | 360 |
| DEFINE DEVCLASS (Define a device class) . . . . .  | 139 | DELETE DRIVE (Delete a drive from a library) . . . . .                                 | 361 |
| DEFINE DOMAIN (Define a new policy domain) . . . . .   | 214 | DELETE EVENT (Delete event records). . . . .   | 362 |
| DEFINE DRIVE (Define a drive to a library) . . . . .   | 216 | DELETE EVENTSERVER (Delete the definition of the event server) . . . . .               | 364 |
| DEFINE EVENTSERVER (Define a server as the event server) . . . . .                                       | 220 | DELETE FILESPACE (Delete client node data from the server) . . . . .                   | 365 |
| DEFINE GRPMEMBER (Add a server to a server group) . . . . .  | 221 | DELETE GRPMEMBER (Delete a server from a server group) . . . . .                       | 369 |
| DEFINE LIBRARY (Define a library). . . . .   | 222 | DELETE KEYRING (Delete password information in the certificate key database) . . . . . | 370 |
| DEFINE MACHINE (Define machine information for disaster recovery) . . . . .                              | 231 | DELETE LIBRARY (Delete a library). . . . .   | 371 |

|  |     |  |     |
|--|-----|--|-----|
| DELETE MACHINE (Delete machine information) . . . . .                                    | 372 | EXPORT POLICY (Export policy information)  | 453 |
| DELETE MACHNODEASSOCIATION (Delete association between a machine and a node) . . . . .   | 373 | EXPORT SERVER (Export server information)  | 459 |
| DELETE MGMTCLASS (Delete a management class) . . . . .                                   | 374 | EXTEND DBSPACE (Increase space for the database) . . . . .                                     | 474 |
| DELETE NODEGROUP (Delete a node group)   | 375 | GENERATE commands . . . . .  | 475 |
| DELETE NODEGROUPMEMBER (Delete node group member). . . . .                               | 376 | GENERATE BACKUPSET (Generate a backup set of a client's data) . . . . .                        | 476 |
| DELETE PATH (Delete a path) . . . . .  | 377 | GENERATE BACKUPSETTOC (Generate a table of contents for a backup set) . . . . .                | 484 |
| DELETE POLICYSET (Delete a policy set) . . . . .   | 379 | GRANT commands . . . . .   | 487 |
| DELETE PROFASSOCIATION (Delete a profile association) . . . . .                          | 380 | GRANT AUTHORITY (Add administrator authority) . . . . .  | 488 |
| DELETE PROFILE (Delete a profile). . . . .   | 383 | GRANT PROXYNODE (Grant proxy authority to a client node) . . . . .                             | 492 |
| DELETE RECMEDMACHASSOCIATION (Delete recovery media and machine association) . . . . .   | 385 | HALT (Shut down the server) . . . . .  | 493 |
| DELETE RECOVERYMEDIA (Delete recovery media) . . . . .                                   | 386 | HELP (Get help on commands and error messages)   | 495 |
| DELETE SCHEDULE (Delete a client or an administrative command schedule) . . . . .        | 387 | IDENTIFY DUPLICATES (Identify duplicate data in a storage pool) . . . . .                      | 497 |
| DELETE SCRIPT (Delete command lines from a script or delete the entire script) . . . . . | 390 | IMPORT commands . . . . .  | 501 |
| DELETE SERVER (Delete a server definition)   | 391 | IMPORT ADMIN (Import administrator information). . . . .                                       | 502 |
| DELETE SERVERGROUP (Delete a server group). . . . .                                      | 392 | IMPORT NODE (Import client node information). . . . .  | 505 |
| DELETE SPACETRIGGER (Delete the storage pool space triggers) . . . . .                   | 393 | IMPORT POLICY (Import policy information)  | 511 |
| DELETE STGPOOL (Delete a storage pool) . . . . .   | 394 | IMPORT SERVER (Import server information)  | 514 |
| DELETE SUBSCRIBER (Delete subscriptions from a configuration manager database) . . . . . | 395 | INSERT MACHINE (Insert machine characteristics information or recovery instructions) . . . . . | 519 |
| DELETE SUBSCRIPTION (Delete a profile subscription) . . . . .                            | 396 | ISSUE MESSAGE (Issue a message from a server script) . . . . .                                 | 521 |
| DELETE VIRTUALFSMAPPING (Delete a virtual file space mapping) . . . . .                  | 397 | LABEL LIBVOLUME (Label a library volume) . . . . .   | 523 |
| DELETE VOLHISTORY (Delete sequential volume history information) . . . . .               | 398 | LOCK commands . . . . .  | 529 |
| DELETE VOLUME (Delete a storage pool volume) . . . . .                                   | 403 | LOCK ADMIN (Lock out an administrator) . . . . .   | 530 |
| DISABLE commands . . . . .   | 406 | LOCK NODE (Lock out a client node) . . . . .   | 531 |
| DISABLE EVENTS (Disable events for event logging) . . . . .                              | 407 | LOCK PROFILE (Lock a profile) . . . . .  | 532 |
| DISABLE SESSIONS (Temporarily prevent client node access to the server) . . . . .        | 411 | MACRO (Invoke a macro) . . . . .   | 534 |
| DISMOUNT command . . . . .   | 413 | MIGRATE STGPOOL (Migrate storage pool to next storage pool) . . . . .                          | 536 |
| DISMOUNT VOLUME (Dismount a volume by volume name) . . . . .                             | 414 | MOVE commands. . . . .   | 539 |
| DISPLAY OBJNAME (Display a full object name)   | 415 | MOVE DATA (Move files on a storage pool volume) . . . . .                                      | 540 |
| ENABLE commands . . . . .  | 416 | MOVE DRMEDIA (Move disaster recovery media offsite and back onsite). . . . .                   | 544 |
| ENABLE EVENTS (Enable server or client events for logging) . . . . .                     | 417 | MOVE GRPMEMBER (Move a server group member) . . . . .  | 558 |
| ENABLE SESSIONS (Resume user activity on the server) . . . . .                           | 420 | MOVE MEDIA (Move sequential access storage pool media) . . . . .                               | 559 |
| END EVENTLOGGING (Stop logging events) . . . . .   | 422 | MOVE NODEDATA (Move data by node in a sequential access storage pool) . . . . .                | 567 |
| EXPIRE INVENTORY (Manually start inventory expiration processing) . . . . .              | 424 | NOTIFY SUBSCRIBERS (Notify managed servers to update profiles). . . . .                        | 576 |
| EXPORT commands . . . . .  | 428 | PARALLEL (Run multiple commands in a script in parallel) . . . . .                             | 577 |
| EXPORT ADMIN (Export administrator information). . . . .                                 | 429 | PING SERVER (Test the connection between servers) . . . . .                                    | 579 |
| EXPORT NODE (Export client node information). . . . .                                    | 435 | PREPARE (Create a recovery plan file) . . . . .  | 580 |
|  |     | QUERY commands . . . . .   | 585 |
|  |     | QUERY ACTLOG (Query the activity log). . . . .   | 587 |
|  |     | QUERY ADMIN (Display administrator information). . . . .                                       | 592 |
|  |     | QUERY ASSOCIATION (Query client node associations with a schedule) . . . . .                   | 596 |



|   |     |
|---|-----|
| QUERY AUDIT OCCUPANCY (Query client node storage utilization).          | 598 |
| QUERY BACKUPSET (Query a backup set)                                    | 601 |
| QUERY BACKUPSETCONTENTS (Query contents of a backup set)                | 606 |
| QUERY CLOPTSET (Query a client option set)                              | 609 |
| QUERY COLLOCGROUP (Query a collocation group).                          | 611 |
| QUERY CONTENT (Query the contents of a storage pool volume).            | 614 |
| QUERY COPYGROUP (Query copy groups)                                     | 621 |
| QUERY DATAMOVER (Display data mover definitions)                        | 626 |
| QUERY DB (Display database information)                                 | 629 |
| QUERY DBSPACE (Display database storage space)                          | 632 |
| QUERY DEVCLASS (Display information on one or more device classes).     | 633 |
| QUERY DIRSPACE (Query storage utilization of FILE directories)          | 637 |
| QUERY DOMAIN (Query a policy domain)                                    | 638 |
| QUERY DRIVE (Query information about a drive)                           | 641 |
| QUERY DRMEDIA (Query disaster recovery media)                           | 645 |
| QUERY DRMSTATUS (Query disaster recovery manager system parameters)     | 654 |
| QUERY ENABLED (Query enabled events)                                    | 657 |
| QUERY EVENT (Query scheduled and completed events)                      | 659 |
| QUERY EVENTRULES (Query rules for server or client events)              | 671 |
| QUERY EVENTSERVER (Query the event server).                             | 673 |
| QUERY EXPORT (Query for active or suspended export operations)          | 674 |
| QUERY FILESPACE (Query one or more file spaces)                         | 681 |
| QUERY LIBRARY (Query a library)   | 686 |
| QUERY LIBVOLUME (Query a library volume)                                | 689 |
| QUERY LICENSE (Display license information)                             | 692 |
| QUERY LOG (Display information on the recovery log)                     | 693 |
| QUERY MACHINE (Query machine information).                              | 695 |
| QUERY MEDIA (Query sequential access storage pool media)                | 698 |
| QUERY MGMTCLASS (Query a management class)                              | 704 |
| QUERY MOUNT (Display information on mounted sequential access volumes). | 707 |
| QUERY NASBACKUP (Query NAS backup images)                               | 708 |
| QUERY NODE (Query nodes).   | 712 |
| QUERY NODedata (Query client data in volumes)                           | 719 |
| QUERY NODEGROUP (Query a node group)                                    | 722 |
| QUERY OCCUPANCY (Query client file spaces in storage pools)             | 724 |
| QUERY OPTION (Query server options)                                     | 728 |
| QUERY PATH (Display a path definition)                                  | 729 |

|  |     |
|--|-----|
| QUERY POLICYSET (Query a policy set)   | 733 |
| QUERY PROCESS (Query one or more server processes)                             | 736 |
| QUERY PROFILE (Query a profile)  | 738 |
| QUERY PROXYNODE (Query proxy authority for a client node)                      | 741 |
| QUERY RECOVERYMEDIA (Query recovery media)                                     | 742 |
| QUERY REQUEST (Query one or more pending mount requests)                       | 745 |
| QUERY RESTORE (Query restartable restore sessions)                             | 746 |
| QUERY RPFCONTENT (Query recovery plan file contents stored on a target server) | 749 |
| QUERY RPFfile (Query recovery plan file information stored on a target server) | 751 |
| QUERY SAN (Query the devices on the SAN)                                       | 754 |
| QUERY SCHEDULE (Query schedules).  | 757 |
| QUERY SCRIPT (Query Tivoli Storage Manager scripts)                            | 765 |
| QUERY SERVER (Query a server)  | 768 |
| QUERY SERVERGROUP (Query a server group)                                       | 772 |
| QUERY SESSION (Query client sessions)  | 774 |
| QUERY SHREDSTATUS (Query shredding status)                                     | 778 |
| QUERY SPACETRIGGER (Query the space triggers)                                  | 780 |
| QUERY STATUS (Query system parameters)   | 782 |
| QUERY STGPOOL (Query storage pools)  | 788 |
| QUERY SUBSCRIBER (Display subscriber information).                             | 799 |
| QUERY SUBSCRIPTION (Display subscription information).                         | 801 |
| QUERY SYSTEM (Query the system configuration and capacity).                    | 803 |
| QUERY TAPEALERTMSG (Display status of SET TAPEALERTMSG command)                | 805 |
| QUERY TOC (Display table of contents for a backup image)                       | 806 |
| QUERY VIRTUALFSMAPPING (Query a virtual file space mapping)                    | 809 |
| QUERY VOLHISTORY (Display sequential volume history information)               | 811 |
| QUERY VOLUME (Query storage pool volumes)                                      | 818 |
| QUIT (End the interactive mode of the administrative client)                   | 825 |
| RECLAIM STGPOOL (Reclaim volumes in a sequential-access storage pool)          | 826 |
| RECONCILE VOLUMES (Reconcile differences in the virtual volume definitions)    | 829 |
| REGISTER commands  | 832 |
| REGISTER ADMIN (Register an administrator)                                     | 833 |
| REGISTER LICENSE (Register a new license)                                      | 835 |
| REGISTER NODE (Register a node)  | 837 |
| REMOVE commands  | 847 |
| REMOVE ADMIN (Delete an administrator)   | 848 |
| REMOVE NODE (Delete a node or an associated machine node)                      | 849 |
| RENAME commands  | 851 |
| RENAME ADMIN (Rename an administrator)   | 852 |



|  |     |   |     |
|--|-----|---|-----|
| RENAME FILESPACE (Rename a client file space on the server) . . . . .                          | 853 | SET DRMCOPYSTGPOOL (Specify the copy storage pools to be managed by DRM) . . . . .        | 917 |
| RENAME NODE (Rename a node) . . . . .  | 856 | SET DRMCOURIERNAME (Specify the courier name) . . . . .                                   | 918 |
| RENAME SCRIPT (Rename a Tivoli Storage Manager script) . . . . .                               | 857 | SET DRMDBBACKUPEXPIREDDAYS (Specify DB backup series expiration) . . . . .                | 919 |
| RENAME SERVERGROUP (Rename a server group). . . . .  | 858 | SET DRMFILEPROCESS (Specify file processing) . . . . .                                    | 921 |
| RENAME STGPOOL (Change the name of a storage pool) . . . . .                                   | 859 | SET DRMINSTRPREFIX (Specify the prefix for recovery instructions file names) . . . . .    | 922 |
| REPLY (Allow a request to continue processing)   | 860 | SET DRMNOTMOUNTABLENAME (Specify the not mountable location name). . . . .                | 924 |
| RESET PASSEXP (Reset password expiration) . . . . .  | 861 | SET DRMPLANPREFIX (Specify a prefix for recovery plan file names) . . . . .               | 925 |
| RESTART EXPORT (Restart a suspended export operation) . . . . .                                | 863 | SET DRMPLANVPOSTFIX (Specify replacement volume names). . . . .                           | 927 |
| RESTORE commands. . . . .  | 865 | SET DRMPRIMSTGPOOL (Specify the primary storage pools to be managed by DRM) . . . . .     | 928 |
| RESTORE NODE (Restore a NAS node) . . . . .  | 866 | SET DRMRPFEXPIREDDAYS (Set criteria for recovery plan file expiration) . . . . .          | 929 |
| RESTORE STGPOOL (Restore storage pool data from a copy pool or an active-data pool) . . . . .  | 871 | SET DRMVaultNAME (Specify the vault name) . . . . .                                       | 931 |
| RESTORE VOLUME (Restore primary volume data from a copy pool or an active-data pool) . . . . . | 875 | SET EVENTRETENTION (Set the retention period for event records) . . . . .                 | 932 |
| REVOKE commands . . . . .  | 880 | SET INVALIDPWLIMIT (Set the number of invalid logon attempts) . . . . .                   | 933 |
| REVOKE AUTHORITY (Remove administrator authority) . . . . .                                    | 881 | SET LICENSEAUDITPERIOD (Set license audit period) . . . . .                               | 934 |
| REVOKE PROXYNODE (Revoke proxy authority for a client node). . . . .                           | 885 | SET MAXCMDRETRIES (Set the maximum number of command retries) . . . . .                   | 935 |
| ROLLBACK (Rollback uncommitted changes in a macro). . . . .                                    | 886 | SET MAXSCHEDULESESSIONS (Set maximum scheduled sessions) . . . . .                        | 936 |
| RUN (Run a Tivoli Storage Manager script) . . . . .  | 887 | SET MINPWLENGTH (Set minimum password length) . . . . .                                   | 937 |
| SELECT (Perform an SQL query of the IBM Tivoli Storage Manager database). . . . .              | 890 | SET PASSEXP (Set password expiration date) . . . . .                                      | 938 |
| SERIAL (Run multiple commands in a script in serial) . . . . .                                 | 897 | SET QUERYSCHEDPERIOD (Set query period for polling client nodes) . . . . .                | 940 |
| SET commands. . . . .  | 898 | SET RANDOMIZE (Set randomization of scheduled start times) . . . . .                      | 941 |
| SET ACCOUNTING (Set accounting records on or off) . . . . .                                    | 900 | SET REGISTRATION (Set open or closed registration) . . . . .                              | 943 |
| SET ACTLOGRETENTION (Set the retention period or the size of the activity log) . . . . .       | 901 | SET RETRYPERIOD (Set time between retry attempts) . . . . .                               | 945 |
| SET ARCHIVERETENTIONPROTECTION (Activate data retention protection) . . . . .                  | 903 | SET SCHEDMODES (Select a central scheduling mode). . . . .                                | 946 |
| SET AUTHENTICATION (Set password authentication) . . . . .                                     | 905 | SET SERVERHLADDRESS (Set the high-level address of a server) . . . . .                    | 948 |
| SET CLIENTACTDURATION (Set the duration period for the client action). . . . .                 | 906 | SET SERVERLLADDRESS (Set the low-level address of a server) . . . . .                     | 949 |
| SET CONFIGMANAGER (Specify a configuration manager) . . . . .                                  | 907 | SET SERVERNAME (Specify the server name) . . . . .  | 950 |
| SET CONFIGREFRESH (Set managed server configuration refresh). . . . .                          | 908 | SET SERVERPASSWORD (Set password for server). . . . .                                     | 951 |
| SET CONTEXTMESSAGING (Set message context reporting on or off). . . . .                        | 909 | SET SUBFILE (Set subfile backup for client nodes). . . . .                                | 952 |
| SET CROSSDEFINE (Specifies whether to cross-define servers) . . . . .                          | 910 | SET SUMMARYRETENTION (Set number of days to keep data in activity summary table). . . . . | 953 |
| SET DBRECOVERY (Set the device class for automatic backups) . . . . .                          | 911 | SET TAPEALERTMSG (Set tape alert messages on or off). . . . .                             | 954 |
| SET DBREPORTMODE (Set the level of database reporting) . . . . .                               | 912 | SET TOCLOADRETENTION (Set load retention period for table of contents) . . . . .          | 955 |
| SET DRMACTIVEDATASTGPOOL (Specify the active-data pools to be managed by DRM) . . . . .        | 913 | SETOPT (Set a server option for dynamic update) . . . . .                                 | 956 |
| SET DRMCHECKLABEL (Specify label checking). . . . .  | 915 |   |     |
| SET DRMCMDFILENAME (Specify the name of a file to contain commands) . . . . .                  | 916 |   |     |

|   |      |
|---|------|
| SHRED DATA (Shred data) . . . . .   | 958  |
| SUSPEND EXPORT (Suspend a currently running export operation) . . . . .               | 960  |
| UNLOCK commands . . . . .   | 961  |
| UNLOCK ADMIN (Unlock an administrator) . . . . .                                      | 962  |
| UNLOCK NODE (Unlock a client node) . . . . .  | 963  |
| UNLOCK PROFILE (Unlock a profile) . . . . .   | 964  |
| UPDATE commands . . . . .   | 965  |
| UPDATE ADMIN (Update an administrator) . . . . .                                      | 966  |
| UPDATE BACKUPSET (Update a retention value assigned to a backup set) . . . . .        | 968  |
| UPDATE CLIENTOPT (Update a client option sequence number) . . . . .                   | 973  |
| UPDATE CLOPTSET (Update a client option set description) . . . . .                    | 974  |
| UPDATE COLLOCGROUP (Update a collocation group) . . . . .                             | 975  |
| UPDATE COPYGROUP (Update a copy group) . . . . .                                      | 976  |
| UPDATE DATAMOVER (Update a data mover) . . . . .                                      | 984  |
| UPDATE DEVCLASS (Update the attributes of a device class) . . . . .                   | 986  |
| UPDATE DOMAIN (Update a policy domain) . . . . .                                      | 1053 |
| UPDATE DRIVE (Update a drive) . . . . .   | 1055 |
| UPDATE LIBRARY (Update a library) . . . . .   | 1059 |
| UPDATE LIBVOLUME (Change the status of a storage volume) . . . . .                    | 1064 |
| UPDATE MACHINE (Update machine information) . . . . .                                 | 1066 |
| UPDATE MGMTCLASS (Update a management class) . . . . .                                | 1068 |
| UPDATE NODE (Update node attributes) . . . . .  | 1071 |
| UPDATE NODEGROUP (Update a node group) . . . . .                                      | 1079 |
| UPDATE PATH (Change a path) . . . . .   | 1080 |
| UPDATE POLICYSET (Update a policy set description) . . . . .                          | 1085 |
| UPDATE PROFILE (Update a profile description) . . . . .                               | 1087 |
| UPDATE RECOVERYMEDIA (Update recovery media) . . . . .                                | 1088 |
| UPDATE SCHEDULE (Update a schedule) . . . . .   | 1090 |
| UPDATE SCRIPT (Update a Tivoli Storage Manager script) . . . . .                      | 1110 |
| UPDATE SERVER (Update a server defined for server-to-server communications) . . . . . | 1112 |
| UPDATE SERVERGROUP (Update a server group description) . . . . .                      | 1116 |
| UPDATE SPACETRIGGER (Update the space triggers) . . . . .                             | 1117 |
| UPDATE STGPOOL (Update a storage pool) . . . . .                                      | 1119 |
| UPDATE VIRTUALFSMAPPING (Update a virtual file space mapping) . . . . .               | 1153 |
| UPDATE VOLHISTORY (Update sequential volume history information) . . . . .            | 1155 |
| UPDATE VOLUME (Change a storage pool volume) . . . . .                                | 1157 |
| VALIDATE commands . . . . .   | 1161 |
| VALIDATE LANFREE (Validate LAN-Free paths) . . . . .                                  | 1162 |
| VALIDATE POLICYSET (Verify a policy set) . . . . .                                    | 1164 |

|   |      |
|---|------|
| VARY (Bring a random access volume online or offline) . . . . . | 1166 |
|---|------|

## Chapter 3. Server options . . . . . 1169

|  |      |
|--|------|
| Modifying server options . . . . .                 | 1169 |
| Types of server options . . . . .                  | 1170 |
| Communication options . . . . .                    | 1170 |
| Server storage options . . . . .                   | 1171 |
| Client-server options . . . . .                    | 1171 |
| Date, number, time, and language options . . . . . | 1172 |
| Database options . . . . .                         | 1172 |
| Data transfer options . . . . .                    | 1172 |
| Message options . . . . .                          | 1173 |
| Event logging options . . . . .                    | 1173 |
| Security options and licensing options . . . . .   | 1173 |
| Miscellaneous options . . . . .                    | 1174 |
| 3494SHARED . . . . .                               | 1175 |
| ACSACCESSID . . . . .                              | 1176 |
| ACSLOCKDRIVE . . . . .                             | 1177 |
| ACSQUICKINIT . . . . .                             | 1178 |
| ACSTIMEOUTX . . . . .                              | 1179 |
| ACTIVELOGDIR . . . . .                             | 1180 |
| ACTIVELOGSIZE . . . . .                            | 1181 |
| ADMINONCLIENTPORT . . . . .                        | 1182 |
| AIXASYNCIO . . . . .                               | 1183 |
| ALIASHALT . . . . .                                | 1184 |
| ARCHFAILOVERLOGDIR . . . . .                       | 1185 |
| ARCHIVELOGDIR . . . . .                            | 1186 |
| ASSISTVCRRECOVERY . . . . .                        | 1187 |
| AUDITSTORAGE . . . . .                             | 1188 |
| COMMETHOD . . . . .                                | 1189 |
| COMMTIMEOUT . . . . .                              | 1190 |
| DBMEMPERCENT . . . . .                             | 1191 |
| DEDUPREQUIRESBACKUP . . . . .                      | 1192 |
| DEVCONFIG . . . . .                                | 1193 |
| DISABLESCHEDULES . . . . .                         | 1194 |
| DISKSTGPPOOLMEMSIZE . . . . .                      | 1194 |
| DISPLAYLFINFO . . . . .                            | 1195 |
| DNSLOOKUP . . . . .                                | 1196 |
| DRIVEACQUIRERETRY . . . . .                        | 1197 |
| EVENTSERVER . . . . .                              | 1198 |
| EXPINTERVAL . . . . .                              | 1199 |
| EXPQUIET . . . . .                                 | 1200 |
| FILEEXIT . . . . .                                 | 1201 |
| FILETEXTEXIT . . . . .                             | 1202 |
| IDLETIMEOUT . . . . .                              | 1203 |
| LANGUAGE . . . . .                                 | 1204 |
| MAXSESSIONS . . . . .                              | 1206 |
| MESSAGEFORMAT . . . . .                            | 1207 |
| MIRRORLOGDIR . . . . .                             | 1208 |
| MOVEBATCHSIZE . . . . .                            | 1209 |
| MOVESIZETHRESH . . . . .                           | 1210 |
| MSGINTERVAL . . . . .                              | 1211 |
| NDMPCONTROLPORT . . . . .                          | 1212 |
| NDMPPORTRANGE . . . . .                            | 1213 |
| NDMPPPREFDATAINTERFACE . . . . .                   | 1214 |
| NOPREEMPT . . . . .                                | 1215 |
| NORETRIEVEDATE . . . . .                           | 1216 |
| NUMOPENVOLSAALLOWED . . . . .                      | 1217 |
| QUERYAUTH . . . . .                                | 1219 |
| RECLAIMDELAY . . . . .                             | 1220 |

|                                   |      |
|-----------------------------------|------|
| RECLAIMPERIOD . . . . .           | 1221 |
| REPORTRETRIEVE . . . . .          | 1222 |
| REQSYSAUTHOUTFILE . . . . .       | 1223 |
| RESOURCETIMEOUT . . . . .         | 1224 |
| RESTOREINTERVAL . . . . .         | 1225 |
| RETENTIONEXTENSION . . . . .      | 1226 |
| SANDISCOVERY . . . . .            | 1227 |
| SANDISCOVERYTIMEOUT . . . . .     | 1228 |
| SANREFRESHTIME . . . . .          | 1229 |
| SEARCHMPQUEUE . . . . .           | 1230 |
| SHMPORT . . . . .                 | 1231 |
| SHREDDING . . . . .               | 1232 |
| SNMPHEARTBEATINTERVAL . . . . .   | 1233 |
| SNMPMESSAGECATEGORY . . . . .     | 1234 |
| SNMPSUBAGENT . . . . .            | 1235 |
| SNMPSUBAGENTHOST . . . . .        | 1236 |
| SNMPSUBAGENTPORT . . . . .        | 1237 |
| SSLTCPADMINPORT . . . . .         | 1238 |
| SSLTCPPOINT . . . . .             | 1239 |
| TCPADMINPORT . . . . .            | 1240 |
| TCPBUFSIZE . . . . .              | 1241 |
| TCPNODELAY . . . . .              | 1242 |
| TCPPOINT . . . . .                | 1243 |
| TCPWINDOWSIZE . . . . .           | 1244 |
| TECBEGINEVENTLOGGING . . . . .    | 1245 |
| TECHOST . . . . .                 | 1246 |
| TECPOINT . . . . .                | 1247 |
| TECUTF8EVENT . . . . .            | 1248 |
| THROUGHPUTDATATHRESHOLD . . . . . | 1249 |
| THROUGHPUTTIMETHRESHOLD . . . . . | 1250 |
| TXNGROUPMAX . . . . .             | 1251 |
| UNIQUETDPTCEVENTS . . . . .       | 1252 |
| UNIQUETECEVENTS . . . . .         | 1253 |
| USEREXIT . . . . .                | 1254 |
| VERBCHECK . . . . .               | 1255 |
| VOLUMEHISTORY . . . . .           | 1256 |

## **Chapter 4. Server utilities. . . . . 1257**

|  |      |
|--|------|
| DSMSERV (Start the server) . . . . .   | 1258 |
| The server startup script: rc.dsmserv . . . . .                                      | 1259 |
| DSMSERV DISPLAY DBSPACE (Display information about database storage space) . . . . . | 1260 |
| DSMSERV DISPLAY LOG (Display recovery log information) . . . . .                     | 1261 |
| DSMSERV FORMAT (Format the database and log) . . . . .                               | 1263 |
| DSMSERV INSERTDB (Move a server database into an empty database) . . . . .           | 1265 |
| DSMSERV LOADFORMAT (Format a database) . . . . .                                     | 1267 |
| DSMSERV REMOVEDB (Remove a database) . . . . .                                       | 1269 |
| DSMSERV RESTORE DB (Restore the database) . . . . .                                  | 1270 |
| DSMSERV RESTORE DB (Restore a database to its most current state) . . . . .          | 1271 |
| DSMSERV RESTORE DB (Restore a database to a point-in-time) . . . . .                 | 1273 |
| DSMULOG (Capture a message to a user log file) . . . . .                             | 1276 |

## **Appendix A. Return codes for use in IBM Tivoli Storage Manager scripts . 1277**

## **Appendix B. Accessibility features for Tivoli Storage Manager . . . . . 1279**

|                          |             |
|--------------------------|-------------|
| <b>Notices . . . . .</b> | <b>1281</b> |
| Trademarks . . . . .     | 1283        |

## **Glossary . . . . . 1285**

## **Index . . . . . 1307**



---

## Preface

IBM® Tivoli® Storage Manager is a client/server program that provides storage management solutions to customers in a multi-vendor computer environment. IBM Tivoli Storage Manager provides an automated, centrally scheduled, policy-managed backup, archive, and space-management facility for file servers and workstations.

---

## Who should read this publication

This reference is intended for anyone who is registered as an administrator. A single administrator can manage Tivoli Storage Manager, or several people can share administrative responsibilities.

You should be familiar with the operating system on which the server resides and the communication protocols required for the client/server environment. You also need to understand the storage management practices of your organization, such as how you are currently backing up workstation files and how you are using storage devices.

---

## Publications

Tivoli Storage Manager publications and other related publications are available online.

You can search all publications in the Tivoli Storage Manager Information Center: <http://publib.boulder.ibm.com/infocenter/tsminfo/v6>.

You can download PDF versions of publications from the Tivoli Storage Manager Information Center or from the IBM Publications Center at <http://www.ibm.com/shop/publications/order/>.

You can also order some related publications from the IBM Publications Center Web site. The Web site provides information for ordering publications from countries other than the United States. In the United States, you can order publications by calling 800-879-2755.

## Tivoli Storage Manager publications

Publications are available for the server, storage agent, client, and Data Protection.

*Table 1. Tivoli Storage Manager server publications*

| Publication title   | Order number |
|---|--------------|
| <i>IBM Tivoli Storage Manager Messages</i>                          | GC23-9787    |
| <i>IBM Tivoli Storage Manager Performance Tuning Guide</i>          | GC23-9788    |
| <i>IBM Tivoli Storage Manager Problem Determination Guide</i>       | GC23-9789    |
| <i>IBM Tivoli Storage Manager for AIX Installation Guide</i>        | GC23-9781    |
| <i>IBM Tivoli Storage Manager for AIX Administrator's Guide</i>     | SC23-9769    |
| <i>IBM Tivoli Storage Manager for AIX Administrator's Reference</i> | SC23-9775    |
| <i>IBM Tivoli Storage Manager for HP-UX Installation Guide</i>      | GC23-9782    |

*Table 1. Tivoli Storage Manager server publications (continued)*

| <b>Publication title</b>   | <b>Order number</b> |
|--|---------------------|
| <i>IBM Tivoli Storage Manager for HP-UX Administrator's Guide</i>                              | SC23-9770           |
| <i>IBM Tivoli Storage Manager for HP-UX Administrator's Reference</i>                          | SC23-9776           |
| <i>IBM Tivoli Storage Manager for Linux Installation Guide</i>                                 | GC23-9783           |
| <i>IBM Tivoli Storage Manager for Linux Administrator's Guide</i>                              | SC23-9771           |
| <i>IBM Tivoli Storage Manager for Linux Administrator's Reference</i>                          | SC23-9777           |
| <i>IBM Tivoli Storage Manager for Sun Solaris Installation Guide</i>                           | GC23-9784           |
| <i>IBM Tivoli Storage Manager for Sun Solaris Administrator's Guide</i>                        | SC23-9772           |
| <i>IBM Tivoli Storage Manager for Sun Solaris Administrator's Reference</i>                    | SC23-9778           |
| <i>IBM Tivoli Storage Manager for Windows Installation Guide</i>                               | GC23-9785           |
| <i>IBM Tivoli Storage Manager for Windows Administrator's Guide</i>                            | SC23-9773           |
| <i>IBM Tivoli Storage Manager for Windows Administrator's Reference</i>                        | SC23-9779           |
| <i>IBM Tivoli Storage Manager Server Upgrade Guide</i>   | SC23-9554           |
| <i>IBM Tivoli Storage Manager for System Backup and Recovery Installation and User's Guide</i> | SC32-6543           |

*Table 2. Tivoli Storage Manager storage agent publications*

| <b>Publication title</b>   | <b>Order number</b> |
|--|---------------------|
| <i>IBM Tivoli Storage Manager for SAN for AIX Storage Agent User's Guide</i>         | SC23-9797           |
| <i>IBM Tivoli Storage Manager for SAN for HP-UX Storage Agent User's Guide</i>       | SC23-9798           |
| <i>IBM Tivoli Storage Manager for SAN for Linux Storage Agent User's Guide</i>       | SC23-9799           |
| <i>IBM Tivoli Storage Manager for SAN for Sun Solaris Storage Agent User's Guide</i> | SC23-9800           |
| <i>IBM Tivoli Storage Manager for SAN for Windows Storage Agent User's Guide</i>     | SC23-9553           |

*Table 3. Tivoli Storage Manager client publications*

| <b>Publication title</b>   | <b>Order number</b> |
|--|---------------------|
| <i>IBM Tivoli Storage Manager for UNIX and Linux: Backup-Archive Clients Installation and User's Guide</i> | SC23-9791           |
| <i>IBM Tivoli Storage Manager for Windows: Backup-Archive Clients Installation and User's Guide</i>        | SC23-9792           |
| <i>IBM Tivoli Storage Manager for Space Management for UNIX and Linux: User's Guide</i>                    | SC23-9794           |
| <i>IBM Tivoli Storage Manager for HSM for Windows Administration Guide</i>                                 | SC23-9795           |
| <i>IBM Tivoli Storage Manager Using the Application Program Interface</i>                                  | SC23-9793           |
| <i>Program Directory for IBM Tivoli Storage Manager z/OS Edition Backup-Archive Client</i>                 | GI11-8912           |
| <i>Program Directory for IBM Tivoli Storage Manager z/OS Edition Application Program Interface</i>         | GI11-8911           |



*Table 4. Tivoli Storage Manager Data Protection publications*

| <b>Publication title</b>  | <b>Order number</b> |
|---|---------------------|
| <i>IBM Tivoli Storage Manager for Advanced Copy Services: Data Protection for Snapshot Devices Installation and User's Guide</i>        | SC33-8331           |
| <i>IBM Tivoli Storage Manager for Databases: Data Protection for Microsoft SQL Server Installation and User's Guide</i>                 | SC32-9059           |
| <i>IBM Tivoli Storage Manager for Databases: Data Protection for Oracle for UNIX and Linux Installation and User's Guide</i>            | SC32-9064           |
| <i>IBM Tivoli Storage Manager for Databases: Data Protection for Oracle for Windows Installation and User's Guide</i>                   | SC32-9065           |
| <i>IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for SAP Installation and User's Guide for DB2</i>       | SC33-6341           |
| <i>IBM Tivoli Storage Manager for Enterprise Resource Planning: Data Protection for SAP Installation and User's Guide for Oracle</i>    | SC33-6340           |
| <i>IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for UNIX, Linux, and OS/400® Installation and User's Guide</i> | SC32-9056           |
| <i>IBM Tivoli Storage Manager for Mail: Data Protection for Lotus Domino for Windows Installation and User's Guide</i>                  | SC32-9057           |
| <i>IBM Tivoli Storage Manager for Mail: Data Protection for Microsoft Exchange Server Installation and User's Guide</i>                 | SC23-9796           |
| <i>Program Directory for IBM Tivoli Storage Manager for Mail (Data Protection for Lotus Domino)</i>                                     | GI11-8909           |

## Related hardware publications

The following table lists related IBM hardware products publications.

For additional information on hardware, see the resource library for tape products at <http://www.ibm.com/systems/storage/tape/library.html>.

| <b>Title</b>  | <b>Order Number</b> |
|---|---------------------|
| <i>IBM TotalStorage 3494 Tape Library Introduction and Planning Guide</i> | GA32-0448           |
| <i>IBM TotalStorage 3494 Tape Library Operator Guide</i>                  | GA32-0449           |
| <i>IBM 3490E Model E01 and E11 User's Guide</i>                           | GA32-0298           |
| <i>IBM Tape Device Drivers Installation and User's Guide</i>              | GC27-2130           |
| <i>IBM TotalStorage Enterprise Tape System 3590 Operator Guide</i>        | GA32-0330           |
| <i>IBM TotalStorage Enterprise Tape System 3592 Operator Guide</i>        | GA32-0465           |

## Support information

You can find support information for IBM products from a variety of sources.

## Getting technical training

Information about Tivoli technical training courses is available online.

Go to <http://www.ibm.com/software/tivoli/education/>.

## Searching knowledge bases

If you have a problem with Tivoli Storage Manager, there are several knowledge bases that you can search.

You can begin with the Tivoli Storage Manager Information Center at <http://publib.boulder.ibm.com/infocenter/tsminfo/v6>. From this Web site, you can search all Tivoli Storage Manager publications.

### Searching the Internet

If you cannot find an answer to your question in the Tivoli Storage Manager information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources, go to the support Web site for Tivoli Storage Manager at <http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html>. From there, you can search a variety of resources including:

- IBM technotes
- IBM downloads
- IBM Redbooks®

If you still cannot find the solution to the problem, you can search forums and newsgroups on the Internet for the latest information that might help you resolve your problem. To share your experiences and learn from others in the user community, go to the Tivoli Storage Manager wiki at <http://www.ibm.com/developerworks/wikis/display/tivolistoragemanager/Home>.

### Using IBM Support Assistant

At no additional cost, you can install on any workstation the IBM Support Assistant, a stand-alone application. You can then enhance the application by installing product-specific plug-in modules for the IBM products that you use.

The IBM Support Assistant helps you gather support information when you need to open a problem management record (PMR), which you can then use to track the problem. The product-specific plug-in modules provide you with the following resources:

- Support links
- Education links
- Ability to submit problem management reports

For more information, see the IBM Support Assistant Web site at <http://www.ibm.com/software/support/isa/>.

## Finding product fixes

A product fix to resolve your problem might be available from the IBM Software Support Web site.

You can determine what fixes are available by checking the Web site:

1. Go to the IBM Software Support Web site at <http://www.ibm.com/software/tivoli/products/storage-mgr/product-links.html>.
2. Click the **Support Pages** link for your Tivoli Storage Manager product.
3. Click **Download**, and then click **Fixes by version**.

## Getting e-mail notification of product fixes

You can get notifications about fixes and other news about IBM products.

To receive weekly e-mail notifications about fixes and other news about IBM products, follow these steps:

1. From the support page for any IBM product, click **My support** in the upper-right corner of the page.
2. If you have already registered, skip to the next step. If you have not registered, click **Register** in the upper-right corner of the support page to establish your user ID and password.
3. Sign in to **My support**.
4. On the My support page, click **Edit profiles** in the left navigation pane, and scroll to **Select Mail Preferences**. Select a product family and check the appropriate boxes for the type of information you want.
5. Click **Submit**.
6. For e-mail notification for other products, repeat steps 4 and 5.

## Contacting IBM Software Support

You can contact IBM Software Support if you have an active IBM software maintenance contract and if you are authorized to submit problems to IBM.

Before you contact IBM Software Support, follow these steps:

1. Set up a software maintenance contract.
2. Determine the business impact of your problem.
3. Describe your problem and gather background information.

Then see “Submit the problem to IBM Software Support” on page xvii for information on contacting IBM Software Support.

## Setting up a software maintenance contract

Set up a software maintenance contract. The type of contract that you need depends on the type of product you have.

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as IBM DB2® and IBM WebSphere® products that run on Microsoft® Windows® or UNIX® operating systems), enroll in IBM Passport Advantage® in one of the following ways:
  - **Online:** Go to the Passport Advantage Web page at <http://www.ibm.com/software/lotus/passportadvantage/>, click **How to enroll**, and follow the instructions.
  - **By Phone:** For the phone number to call in your country, go to the IBM Software Support Handbook Web page at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html> and click **Contacts**.

- For server software products, you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for server software products, go to the IBM Technical support advantage Web page at <http://www.ibm.com/servers/>.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. For a list of telephone numbers of people who provide support for your location, go to the Software Support Handbook page at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html>.

### Determine the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting.

|                   |  |
|-------------------|--|
| <b>Severity 1</b> | <b>Critical</b> business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution. |
| <b>Severity 2</b> | <b>Significant</b> business impact: The program is usable but is severely limited.   |
| <b>Severity 3</b> | <b>Some</b> business impact: The program is usable with less significant features (not critical to operations) unavailable.                                      |
| <b>Severity 4</b> | <b>Minimal</b> business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.               |

### Describe the problem and gather background information

When explaining a problem to IBM, it is helpful to be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently.

To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can the problem be recreated? If so, what steps led to the failure?
- Have any changes been made to the system? For example, hardware, operating system, networking software, and so on.
- Are you currently using a workaround for this problem? If so, be prepared to explain it when you report the problem.

## Submit the problem to IBM Software Support

You can submit the problem to IBM Software Support online or by phone.

### Online

Go to the IBM Software Support Web site at <http://www.ibm.com/software/support/probsub.html>. Enter your information into the appropriate problem submission tool.

### By phone

For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook at <http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html>.

If the problem that you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. If a workaround is possible, IBM Software Support provides one for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Tivoli Storage Manager product support Web site at <http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html>, so that users who experience the same problem can benefit from the same resolutions.

---

## Conventions used in this publication

- Command to be entered on the AIX<sup>®</sup> command line:  
`> dsmadm`
- Command to be entered on the command line of an administrative client:  
`query devclass`

In the usage and descriptions for administrative commands, the term characters corresponds to the number of bytes available to store an item. For languages in which it takes a single byte to represent a displayable character, the character to byte ratio is 1 to 1. However, for DBCS and other multi-byte languages, the reference to characters refers only to the number of bytes available for the item and may represent fewer actual characters.





---

# New for IBM Tivoli Storage Manager Version 6.1

Many features in the Tivoli Storage Manager Version 6.1 server are new for previous Tivoli Storage Manager users.

---

## New for the server in Version 6.1.2

Server fix pack 6.1.2 contains several new features, in addition to fixes for problems.

### Enabled functions

Functions that were disabled in Tivoli Storage Manager V6.1.0 and V6.1.1 are now enabled in Version 6.1.2.

Until Tivoli Storage Manager V6.1.2, a database that contained backup sets or tables of contents (TOCs) could not be upgraded to V6. These restrictions no longer exist.

In addition, the following commands have been enabled in Version 6.1.2:

- BACKUP NAS client command if the TOC parameter specifies PREFERRED or YES
- BACKUP NODE if the TOC parameter specifies PREFERRED or YES
- DEFINE BACKUPSET
- GENERATE BACKUPSET
- GENERATE BACKUPSETTOC

### Licensing changes

Following the release of Tivoli Storage Manager Version 6.1.2, Tivoli Storage Manager Version 6.1.0 will no longer be available for download or purchase. Due to this unique circumstance, certain 6.1.2 packages will be available with a license module. See the following information for details on how this situation affects your environment.

#### Existing Version 6.1.0 and 6.1.1 users

If you have installed version 6.1.0 and are using a version 6.1.0 license, you can download the 6.1.2 package from the Service FTP site. You can install the 6.1.2 package using the instructions in Installing a Tivoli Storage Manager fix pack.

#### Version 5 users

If you have not yet installed a version of the V6.1 server, when you upgrade, you must upgrade directly to version 6.1.2. Version 6.1.2 is available with a license module from Passport Advantage or from your Tivoli Storage Manager sales representative. You can upgrade from V5 to V6.1.2 using the instructions in Upgrading the server.

#### New users

Version 6.1.2 is available from Passport Advantage or from your Tivoli Storage Manager sales representative. You can install version 6.1.2 using the instructions in Installing Tivoli Storage Manager.

### TSMDLST utility for AIX

The tsmdlst utility is now available for AIX beginning in Tivoli Storage Manager fix pack level 6.1.2.

Use the tsmdlst utility to obtain information for AIX operating systems about medium-changer, tape, and optical devices controlled by the Tivoli Storage Manager device driver. With this utility you can obtain usage information, device names, serial numbers, and other device information.

### Device support for AIX

The number of devices per driver is extended for AIX.

The Tivoli Storage Manager device driver can configure 1024 devices for each driver.

### PREVIEW parameter for DSMSERV INSERTDB

A **PREVIEW** parameter is available for the DSMSERV INSERTDB utility in Tivoli Storage Manager fix pack level 6.1.2. The DSMSERV INSERTDB utility is used only as part of the process for upgrading a V5 Tivoli Storage Manager server to V6.1.

When you use the **PREVIEW=YES** parameter, the operation includes all the steps of the process, except for the actual insertion of data into the new database.

When you preview the insertion operation, you can quickly verify that the source database is readable. You can also identify any data constraint violations before you run the actual upgrade process for your server.

### New server option SANDISCOVERYTIMEOUT

A new server option is available for SAN discovery functions beginning in Tivoli Storage Manager fix pack level 6.1.2.

The SANDISCOVERYTIMEOUT option specifies the amount of time allowed for host bus adapters to respond when they are queried by the SAN discovery process.

#### Related reference

“SANDISCOVERYTIMEOUT” on page 1228

The SANDISCOVERYTIMEOUT option specifies the amount of time allowed for host bus adapters to respond when they are queried by the SAN discovery process. Once the time specified for the SANDISCOVERYTIMEOUT is reached, the process times out.

---

## New for the server in Version 6.1.0

Tivoli Storage Manager server version 6.1.0 contains many new features and changes.

## Disabled functions in 6.1.0 and 6.1.1

Some functions have been disabled in Tivoli Storage Manager 6.1.0 and 6.1.1.

**Note:** The restrictions described here have been removed in Tivoli Storage Manager V6.1.2. If your server is at the V6.1.0 or V6.1.1 level, migrate to V6.1.2 to enable these functions.

A database containing backup sets or tables of contents (TOCs) cannot be upgraded to V6.1.0 or 6.1.1. The database upgrade utilities check for defined backup sets and existing TOCs. If either exists, the upgrade stops and a message is issued saying that the upgrade is not possible at the time. In addition, any operation on a V6.1 server that tries to create or load a TOC fails.

When support is restored by a future V6.1 fix pack, the database upgrade and all backup set and TOC operations will be fully enabled.

In the meantime, the following commands have been disabled:

- BACKUP NAS client command if the TOC parameter specifies PREFERRED or YES
- BACKUP NODE if the TOC parameter specifies PREFERRED or YES
- DEFINE BACKUPSET
- GENERATE BACKUPSET
- GENERATE BACKUPSETTOC

## Changes to the Version 6.1 Administration Center

Many features in the Tivoli Storage Manager Administration Center Version 6.1 are new for previous users.

### Updated Integrated Solutions Console

In V6.1, the Administration Center is hosted by the IBM Integrated Solutions Console (ISC) Advanced Edition Version 7.1. After installation of the Integrated Solutions Console installation completes, open a Web browser and enter the following URL, which will display the logon screen for the Integrated Solutions Console: [https://local\\_host:9043/ibm/console](https://local_host:9043/ibm/console). This screen indicates a successful installation of the Integrated Solutions Console.

To learn about console updates:

1. Start the ISC.
2. Click **Help** in the ISC banner.
3. In the Help navigation tree, click **Console Updates**.

### Identify managing servers

The table of servers that is the hub of the enterprise-management work page has a column that identifies the managing server, if one exists, for each listed server. By sorting or filtering on the column, you can display the set of servers that are managed by a given server.

### Hover help for table links

The Administration Center typically displays Tivoli Storage Manager objects in a table. In V6.1, when the cursor hovers over an object image, hover-help text is

## Functions enabled in the V6.1.2 server

displayed. The hover help identifies the default action that results when you click the link that is associated with the object.

### Links to information about server messages and Administration Center messages

When a problem or issue occurs with the server or Administration Center, you are immediately notified and provided with a brief message about the problem or issue. The message number is also provided. In V6.1, you can obtain detailed information about a message by clicking the link that is associated with the message number. The information is displayed in a new browser window.

### Maintenance script enhancements

Tivoli Storage Manager utilizes a maintenance script to perform scheduled maintenance tasks. In V6.1, you can generate a maintenance script in one of two styles: predefined and custom.

A *predefined* maintenance script is one that is generated through a wizard. This script contains standard commands that cannot be altered. A predefined script can only be modified in the wizard.

A *custom* maintenance script is created using the Administration Center maintenance script editor. To have more control of your maintenance tasks, you can modify the commands that you specify. You can also use the editor to update your custom maintenance script.

### Client nodes and backup sets enhancements

The redesigned Administration Center displays information about backup sets, client nodes, and client-node groups in one portlet. The design includes search functions that you can use to find and display information more quickly. When you select a client node, a summary panel is displayed with the current operation status, server actions, and client-node actions.

The work item **Client nodes and backup sets** appears in the ISC navigation tree.

### Session and process information available in the health monitor

The Administration Center health monitor now includes information about server processes and sessions. The information is also available in the properties notebooks for servers.

### Centralized server-connection management

In V6.1, server-connection tasks, such as adding a server connection, changing a password, and creating a server instance, are consolidated in a single location: the **Manage Servers** work item, located in the ISC navigation tree.

With actions available in this work item, you can quickly upload server-connection information to the Administration Center using an XML file. This file can optionally include a set of server credentials for multiple servers. To help create an XML file, you can download a list of server connections, without the credential information.

## Changes to management-class activation

In the V6.1, Tivoli Storage Manager no longer activates changes to existing management classes automatically. You must activate the changes manually. Before the changes take effect, they are validated. Results of the validation are displayed. You or another administrator can review them, and then either confirm or cancel the activation.

Because changes are manually activated, you can prepare the management class in advance and activate the changes at an appropriate time.

## Data deduplication

Data deduplication is a method of eliminating redundant data in sequential-access disk (FILE) primary, copy, and active-data storage pools. One unique instance of the data is retained on storage media, and redundant data is replaced with a pointer to the unique data copy. The goal of deduplication is to reduce the overall amount of time that is required to retrieve data by letting you store more data on disk, rather than on tape.

Data deduplication in Tivoli Storage Manager is a two-phase process. In the first phase, duplicate data is identified. During the second phase, duplicate data is removed by certain server processes, such as reclamation processing of storage-pool volumes. By default, a duplicate-identification process begins automatically after you define a storage pool for deduplication. (If you specify a duplicate-identification process when you update a storage pool, it also starts automatically.) Because duplication identification requires extra disk I/O and CPU resources, Tivoli Storage Manager lets you control when identification begins as well as the number and duration of processes.

You can deduplicate any type of data except encrypted data. You can deduplicate client backup and archive data, Tivoli Data Protection data, and so on. Tivoli Storage Manager can deduplicate whole files as well as files that are members of an aggregate. You can deduplicate data that has already been stored. No additional backup, archive, or migration is required.

For optimal efficiency when deduplicating, upgrade to the version 6.1 backup-archive client.

**Restriction:** You can use the data-deduplication feature with Tivoli Storage Manager Extended Edition only.

## Functions enabled in the V6.1.2 server

### Related reference

“DEDUPREQUIRESBACKUP” on page 1192

The DEDUPREQUIRESBACKUP option specifies whether volumes in primary sequential-access storage pools that are set up for deduplication can be reclaimed and whether duplicate data can be discarded before the storage pools are backed up.

“DEFINE STGPOOL (Define a primary storage pool assigned to sequential access devices)” on page 302

Use this command to define a primary storage pool assigned to sequential access devices.

“DEFINE STGPOOL (Define a copy storage pool assigned to sequential access devices)” on page 317

Use this command to define a copy storage pool assigned to sequential access devices.

“DEFINE STGPOOL (Define an active-data pool assigned to sequential-access devices)” on page 325

Use this command to define an active-data pool assigned to sequential-access devices.

“IDENTIFY DUPLICATES (Identify duplicate data in a storage pool)” on page 497

Use this command to start or stop processes that identify duplicate data in a storage pool. You can specify the number of duplicate-identification processes and their duration.

“QUERY PROCESS (Query one or more server processes)” on page 736

Use this command to display information on active background processes.

“QUERY STGPOOL (Query storage pools)” on page 788

Use this command to display information about one or more storage pools. You can also use this command to monitor migration processes for the storage pool.

“UPDATE STGPOOL (Update a primary sequential access pool)” on page 1128

Use this command to update a primary sequential access storage pool.

“UPDATE STGPOOL (Update a copy sequential access storage pool)” on page 1141

Use this command to update a copy sequential access storage pool.

“UPDATE STGPOOL (Update an active-data sequential access)” on page 1147

Use this command to update an active-data pool.

## Storage devices

New device support and other changes to storage devices are available in Tivoli Storage Manager Version 6.1.

### Support for HP and Quantum DAT160 drives and media

With Tivoli Storage Manager, you can now use HP and Quantum DAT160 (DDS6) tape drives and media. New recording formats are available for the 4MM device type.

### Related reference

“DEFINE DEVCLASS (Define a 4MM device class)” on page 152

Use the 4MM device class when you are using 4 mm tape devices.

“UPDATE DEVCLASS (Update a 4MM device class)” on page 998

Use the 4MM device class when you are using 4 mm tape devices.



## **Support for Sun StorageTek T10000 drives, T10000B drives, and T10000 media**

With Tivoli Storage Manager, you can now use Sun StorageTek T10000 drives, T10000B drives, and T10000 media. New recording formats are available for the ECARTRIDGE device type. Tivoli Storage Manager supports Volsafe media with the Sun StorageTek T10000 and T10000B drives.

### **Related reference**

“DEFINE DEVCLASS (Define an ECARTRIDGE device class)” on page 177

Use the ECARTRIDGE device class when you are using StorageTek drives such as the StorageTek SD-3, 9490, 9840 or 9940.

“UPDATE DEVCLASS (Update an ECARTRIDGE device class)” on page 1020

Use the ECARTRIDGE device class when you are using StorageTek drives such as the StorageTek SD-3, 9490, 9840 or 9940.

## **Disaster recovery manager support for active-data pools**

To restore your client systems more quickly and efficiently, you can now use active-data pools in your recovery plans and procedures.

Active-data pools are storage pools that contain only active versions of client backup data. Like copy storage pool volumes, disaster recovery manager lets you:

- Specify the names of active-data pool volumes to be managed by the disaster recovery manager.
- Recycle on-site and off-site active-data pool volumes according to server policies and processes.
- Include active-data pool volumes in the scripts, macros, and documentation that is part of the recovery plan file.
- Track and manage active-data pool media as required by your operations.

By default, active-data pools are not eligible for processing at the time of installation. Copy storage pools, on the other hand, are processed at installation time even if you have not explicitly specified a copy storage pool or pools to be managed.

### Related reference

“MOVE DRMEDIA (Move disaster recovery media offsite and back onsite)” on page 544

Use this command to track database backup and copy storage pool volumes that are to be moved offsite and to identify the expired or empty volumes that are to be moved onsite.

“PREPARE (Create a recovery plan file)” on page 580

Use this command to create a recovery plan file, which contains the information needed to recover a Tivoli Storage Manager server. You can store a recovery plan file on a file system that is accessible to the source Tivoli Storage Manager server or on a target Tivoli Storage Manager server.

“QUERY DRMEDIA (Query disaster recovery media)” on page 645

Use this command to display information about database backup and copy storage pool volumes. You can also use the command to create a file of executable commands to process the volumes.

“QUERY DRMSTATUS (Query disaster recovery manager system parameters)” on page 654

Use this command to display information about the system parameters defined for disaster recovery manager (DRM).

“SET DRMACTIVEDATASTGPOOL (Specify the active-data pools to be managed by DRM)” on page 913

Use this command to specify names of the active-data pools to be recovered after a disaster. Tivoli Storage Manager uses these names if the PREPARE , MOVE DRMEDIA, or QUERY DRMEDIA command does not include the ACTIVEDATASTGPOOL parameter.

## EXPIRE INVENTORY command enhancements

The EXPIRE INVENTORY command is now enhanced with new functionality.

The additional parameters that you can now use are NODE, DOMAIN, TYPE, DURATION, AND RESOURCE. You can use these parameters to target specific client nodes and domains, and also to determine the type of data to be processed. You can use the RESOURCE parameter to specify the number of parallel processes that you want to run within the single EXPIRE INVENTORY process. You can run up to ten threads at one time, but if you are processing one node, only one thread is utilized.

### Related reference

“EXPIRE INVENTORY (Manually start inventory expiration processing)” on page 424

Use this command to manually start inventory expiration processing. The inventory expiration process removes client backup and archive file copies from server storage based on policy specified in the backup and archive copy groups of the management classes to which the files are bound.

## No-query restore changes

The no-query restore (NQR) function and the internal algorithms responsible for NQR were changed to take advantage of DB2 capabilities and to improve performance.

The NQR function has been rewritten to resolve a performance problem encountered when restoring a small number of objects for a client file system with a large number of backup objects spread across a large number of Tivoli Storage Manager server storage pool volumes. NQR performance is now comparable to that of the classic restore under these conditions. NQR now performs a volume determination phase that must be completed before any objects are restored from DISK, FILE, or tape storage volumes.

## Server database

Tivoli Storage Manager version 6.1 provides a new server database. Advantages include automatic statistics collection and database reorganization, full-function SQL queries, and elimination of the need for offline audits of the database.

Upgrading to V6.1 requires that data in a current Tivoli Storage Manager server database be extracted and then inserted into the new database structure. Tivoli Storage Manager provides utilities to perform the process.

## Support for NetApp SnapMirror to Tape feature

With Tivoli Storage Manager you can create SnapMirror to Tape images of file systems on NetApp file servers.

SnapMirror to Tape provides an alternative method for backing up very large NetApp file systems. Because this backup method has limitations, use this method when copying very large NetApp file systems to secondary storage for disaster recovery purposes.

### Related reference

“BACKUP NODE (Back up a NAS node)” on page 49

Use this command to start a backup operation for a network-attached storage (NAS) node.

“RESTORE NODE (Restore a NAS node)” on page 866

Use this command to initiate a restore operation for a network-attached storage (NAS) node.

“QUERY NASBACKUP (Query NAS backup images)” on page 708

Use this command to display information about the file system image objects that have been backed up for a specific NAS node and file space. You can only use this command to display objects that were backed up for a NAS node using NDMP.

## Reporting and monitoring feature

The reporting and monitoring feature uses a combination of the Tivoli Common Reporting tool, IBM Tivoli Monitoring, and the IBM Tivoli Data Warehouse to offer you reports and real time monitoring information about Tivoli Storage Manager servers and client activity.

### ODBC driver support

Tivoli Storage Manager Version 6.1 uses the DB2® open database connectivity (ODBC) driver to query the database and display the results.

The Tivoli Storage Manager ODBC driver is no longer supported with the server.

### Backup sets and client node enhancements

The Administration Center now displays the backup sets, client nodes, and client node groups from one portlet.

You can view all of the client nodes, view them by server, or search for a node from the three available client node tabs. The All Client Nodes tab lists all of the nodes and has a Filter feature to help in your search. The filter works differently than the other table filters in the Administration Center, in that here you do not have to press the Enter key to get results. The search is initiated when you enter a text character in the filter field. As you add characters, the results are filtered even more.

When you select a client node, a summary panel is displayed with the current operation status, server actions, and client node actions. You can also access the Server Actions by right-clicking on the selected row.

The Search tab lets you refine your search parameters to include the server name, the client node name, policy domain name, and other fields that are available.

In the Client Node Groups section, you can find a client node group from the All Client Node Groups tab or from the By Server tab. You can use the filter and the right-click menu on these pages also.

Backup sets are found in the Backup Set Collections section. Search a server by selecting it and clicking **Update Table**.

---

## Command and option changes

Use the lists of new, changed, and deleted commands and options to help you identify operation changes that are needed for your server.

---

### New server commands, utilities, and options

New commands, utilities, and options are available for the V6.1 server because of changes in database operations and new functions.

“New commands”

“New utilities”

“New server options” on page xxx

#### New commands

Table 5. New commands

| Command                  | Function   | Comparable commands in previous versions |
|--------------------------|--|--|
| EXTEND DBSPACE           | Makes additional storage space available for the server to use for its database.<br><br>You can have multiple locations for the database storage space. After installation and initial use of DSMSERV FORMAT or DSMSERV LOADFORMAT, you can add more locations for the storage space for the database. | DEFINE DBVOLUME<br><br>EXTEND DB         |
| IDENTIFY DUPLICATES      | Starts or stops processes that identify duplicate data in a storage pool.  | None                                     |
| QUERY DBSPACE            | Displays the current locations for the database storage, along with total space, and used space.   | QUERY DBVOLUME                           |
| SET DBRECOVERY           | Sets the device class to use for backup of the server's database. The database manager uses the device class for automatic backups of the database.  | DEFINE DBBACKUPTRIGGER                   |
| SET DBREPORTMODE         | Sets the amount of diagnostic information that is reported for the database.   | None                                     |
| SET DRMACTIVEDATASTGPOOL | Sets the active-data pools that are included in your recovery plans and procedures.  | None                                     |

#### New utilities

Table 6. New utilities

| Command                 | Function  | Comparable commands in previous versions |
|-------------------------|---|--|
| DSMSERV DISPLAY DBSPACE | Offline utility to view the current locations for database storage.   | DSMSERV DISPLAY DBVOLUMES                |
| DSMSERV DISPLAY LOG     | Offline utility to view information about the recovery logs (active log and archive logs).                                  | DSMSERV DISPLAY LOGVOLUMES               |
| DSMSERV INSERTDB        | Offline utility used only for inserting data that has been extracted from a V5 server database into an empty V6.1 database. | None                                     |

Table 6. New utilities (continued)

| Command             | Function  | Comparable commands in previous versions |
|---------------------|---|--|
| DSMSERV REMOVEDB    | Offline utility used only when you are sure that you no longer need a server's database and recovery logs. Use with caution.  | None                                     |
| DSMUPGRD PREPAREDDB | Offline utility used only on a V5 server to prepare the database for extraction. After this utility is run, the data can be extracted from the database using the DSMUPGRD EXTRACTDB utility. This is one of the upgrade utilities. | None                                     |
| DSMUPGRD EXTRACTDB  | Offline utility used only on a V5 server to extract the data from the database. The extracted data is inserted into a V6.1 database using the DSMSERV INSERTDB utility. This is one of the upgrade utilities.                       | None                                     |
| DSMUPGRD EXTEND DB  | Offline utility used only on a V5 server to extend the database when database space is insufficient to successfully complete the upgrade process. This is one of the upgrade utilities.   | None                                     |
| DSMUPGRD EXTEND LOG | Offline utility used only on a V5 server to extend the recovery log when recovery log space is insufficient to successfully complete the upgrade process. This is one of the upgrade utilities.                                     | None                                     |
| DSMUPGRD QUERYDB    | Offline utility used only on a V5 server to display information about the database and recovery log. This is one of the upgrade utilities.  | None                                     |

## New server options

Table 7. New server options. For some options, changes have no effect until the server is restarted.

| New server option   | Function   | Comparable commands or options in previous versions |
|---------------------|--|---|
| ACTIVELOGDIR        | The new directory for the location where the active log is stored. Use this option to change the location of the active log.<br><br>The location is originally specified during installation. There is only one location for the active log. | DEFINE LOGVOLUME                                    |
| ACTIVELOGSIZE       | The maximum size of the active log.  | EXTEND LOG<br>REDUCE LOG                            |
| ARCHFAILOVERLOGDIR  | The directory in which the server stores archive log files if they cannot be stored in the archive log location.   | None  |
| ARCHIVELOGDIR       | The directory in which the server stores the archive log.  | None  |
| DBMEMPERCENT        | Sets a limit on the percentage of the system memory that is used for the database.   | None  |
| DEDUPREQUIRESBACKUP | Control for backup operations for primary sequential-access storage pools that are set up for deduplication.   | None  |



Table 7. New server options (continued). For some options, changes have no effect until the server is restarted.

| New server option  | Function  | Comparable commands or options in previous versions |
|--|---|---|
| DISKSTGPOLMEMSIZE  | The size of the cache that the server can use to manage operations for storage pools with the device type of DISK.      | None  |
| MIRRORLOGDIR   | The directory where the log mirror for the active log is stored.  | DEFINE LOGCOPY                                      |
| AIX, HP-UX, Linux, and Sun Solaris systems:<br>SANDISCOVERYTIMEOUT | The amount of time that is allowed for host bus adapters to respond when they are queried by the SAN discovery process. | None  |

## Updated server commands, utilities, and options

Commands, utilities, and options are updated for V6.1 because of changes in database operations and other new functions.

“Updated commands”

“Updated utilities” on page xxxii

“Updated server options” on page xxxiii

### Updated commands

Table 8. Updated commands

| Command   | Changes  |
|---|--|
| BACKUP DB   | The SET DBRECOVERY command must be run first to set a device class for database backups.<br><br>An incremental database backup is now a backup of all changes since the last full backup. In earlier versions of the server, an incremental backup was a backup of all changes since either the last full backup or the last incremental backup. |
| BACKUP NODE<br>RESTORE NODE<br>QUERY NASBACKUP  | The commands support creating SnapMirror to Tape images of file systems on NetApp file servers.  |
| BACKUP VOLHISTORY<br>DELETE VOLHISTORY<br>QUERY VOLHISTORY<br>UPDATE VOLHISTORY         | Database dump operations are no longer available, therefore database dump volumes do not appear in the volume history. Query output is changed.  |
| DEFINE DEVCLASS<br>UPDATE DEVCLASS  | Device formats have been added for some operating systems.   |
| DEFINE SPACETRIGGER<br>DELETE SPACETRIGGER<br>QUERY SPACETRIGGER<br>UPDATE SPACETRIGGER | The space trigger commands now support space triggers only for storage pools. The database and log space triggers are no longer available.   |
| DEFINE STGPPOOL<br>QUERY STGPPOOL<br>UPDATE STGPPOOL                                    | The storage pool commands support the data deduplication functions.  |
| DEFINE VOLUME   | The maximum capacity of a volume in a DISK storage pool is 8 TB.   |

Table 8. Updated commands (continued)

| Command   | Changes  |
|---|--|
| EXPIRE INVENTORY  | Expiration can be run for specific nodes and node groups, or for all nodes in a policy domain. The types of data to be examined for expiration can also be specified.  |
| GRANT AUTHORITY<br>REVOKE AUTHORITY                         | <b>ANALYST</b> privilege class is removed.   |
| HALT  | The <b>QUIESCE</b> parameter is no longer needed.  |
| MOVE DRMEDIA<br>PREPARE<br>QUERY DRMEDIA<br>QUERY DRMSTATUS | Changes to disaster recovery manager commands allow you to include active-data pools in your recovery plans and procedures.  |
| QUERY DB  | Output is changed.   |
| QUERY LOG   | Output is changed.   |
| QUERY OPTION  | Obsolete options are removed from the output.  |
| QUERY PROCESS   | Information about duplicate identification processes is available through this command.  |
| QUERY SESSION   | A new field in the output indicates the actions that occurred during the session.  |
| QUERY STATUS  | Output is changed. Obsolete options are removed, and the database backup trigger is removed.   |
| SELECT  | Some parameters that were previously allowed might cause errors. SQL parameter usage must now conform to usage that is acceptable to the database manager. See “Changes to the SELECT command” on page xxxvii. |
| SETOPT  | Obsolete options are removed.  |
| UPDATE ADMIN  | <b>ANALYST</b> privilege class is removed.   |

## Updated utilities

Table 9. Updated utilities

| Utility                       | Changes  |
|-------------------------------|--|
| DSMSERV (starting the server) | New options are available for specifying the owning user ID for the server instance on startup. The new options are also available for other DSMSERV utilities.  |
| DSMSERV FORMAT                | <p>Obsolete parameters are removed. New parameters are added to specify the directories for database space, and the maximum size and locations of the recovery log.</p> <p>This utility is used to format a database for installation of a new server.</p> |

Table 9. Updated utilities (continued)

| Utility  | Changes   |
|--|---|
| DSMSERV LOADFORMAT   | This utility is used only for formatting a new, completely empty database. An empty database is used only as part of the process of upgrading an earlier version of the server to V6.1. After you format an empty database, you use the DSMSERV INSERTDB utility to insert data that was extracted from the database of an earlier version of the server. |
| DSMSERV RESTORE DB <ul style="list-style-type: none"> <li>Restore a database to its most current state</li> <li>Restore a database to a point in time</li> </ul> | <p>Volume history is now required for restoring the database.</p> <p>All restore operations use roll-forward recovery.</p> <p>The function for restoring individual database volumes was removed. The server no longer manages database volumes.</p>  |

## Updated server options

Table 10. Updated server options

| Option      | Changes  |
|-------------|--|
| TXNGROUPMAX | <p>The default value is increased from 256 to 4096. Check whether the server options file has this option:</p> <ul style="list-style-type: none"> <li>If the server options file does not include this option, the server automatically uses the new default value.</li> <li>If the server options file includes a value for the option, the server uses that specified value. If the specified value is less than 4096, consider increasing this value, or removing the option so that the server uses the new default value.</li> </ul> <p>Increasing the value or using the new default value can improve the performance for data movement operations such as storage pool migration and storage pool backup.</p> <p>Increasing the value for the TXNGROUPMAX option has no effect on data-movement performance for files that were stored on the server using a lower value for the option.</p> |

---

## Deleted server commands, utilities, and options

Some commands, utilities, and options are deleted because their function is no longer needed. In some cases, new commands replace deleted commands.

“Deleted commands”

“Deleted utilities” on page xxxv

“Deleted server options” on page xxxvi

### Deleted commands

*Table 11. Deleted commands*

| Deleted command   | Comments   |
|---|--|
| CONVERT ARCHIVE   | The operation that this command performed is no longer needed.   |
| DEFINE DBBACKUPTRIGGER<br>DELETE DBBACKUPTRIGGER<br>QUERY DBBACKUPTRIGGER<br>UPDATE DBBACKUPTRIGGER | Ensure that you schedule backups of the database to occur at least once per day.   |
| DEFINE DBCOPY   | Database volumes are no longer used.   |
| DEFINE DBVOLUME<br>QUERY DBVOLUME   | Space allocation is done automatically in the directory locations specified for the database.  |
| DEFINE LOGCOPY  | Instead of log volume copies, you can specify a log mirror to have the active log protected by a mirror copy.  |
| DEFINE LOGVOLUME<br>DELETE LOGVOLUME<br>QUERY LOGVOLUME   | The database manager automatically manages space in the recovery log directories.<br><br>For information about the directories that are used for the logs, use the QUERY LOG command.                    |
| ESTIMATE DBREORGSTATS   | Collecting and resetting database statistics occurs automatically.<br><br>Database-reorganization operations are done automatically by the database manager as needed.                                   |
| EXTEND DB   | Space allocation is done automatically in the directory locations specified for the database. If the server needs additional space, you can add directory locations by using the EXTEND DBSPACE command. |
| EXTEND LOG  | Server options are available for increasing the size of recovery logs.   |
| QUERY SQLSESSION  | The information that this command supplied is no longer in the server database. SQL SELECT settings are replaced by syntax options that are available in a DB2 SELECT command.                           |
| REDUCE DB   | Space allocation is done automatically in the directory locations specified for the database. You cannot adjust the assigned capacity of the database.   |

Table 11. Deleted commands (continued)

| Deleted command   | Comments  |
|---|---|
| REDUCE LOG  | The database manager automatically manages space in the recovery log directories. |
| RESET BUFPOOL   | The BUFPOOLSIZ option has been eliminated, therefore this command is not needed.  |
| RESET DBMAXUTILIZATION<br>RESET LOGCONSUMPTION<br>RESET LOGMAXUTILIZATION | Collecting and resetting database statistics occurs automatically.                |
| SET LOGMODE   | Logging mode for the database is now always roll-forward mode.                    |
| SET SQLDATETIMEFORMAT<br>SET SQLDISPLAYMODE<br>SET SQLMATHMODE            | The commands are replaced by options in the DB2 SELECT command syntax.            |
| UNDO ARCHCONVERSION   | The operation that this command performed is no longer needed.                    |
| UPDATE ARCHIVE  | The operation that this command performed is no longer needed.                    |

## Deleted utilities

Table 12. Deleted utilities

| Deleted utility                | Comments  |
|--------------------------------|---|
| DSMFMT                         | Space allocation is done automatically in the directory locations specified for the database.   |
| DSMSERV AUDITDB                | Offline database audits are no longer required.<br><br>As data is added to the server database, the database manager automatically checks data constraints and data types. The online integrity checks prevent problems for which offline audits had been needed in earlier releases. |
| DSMSERV DISPLAY DBBACKUPVOLUME | Information about volumes used for database backup is available from the volume history file. The volume history file is now required to restore the database.  |
| DSMSERV DISPLAY DBVOLUMES      | Use DSMSERV DISPLAY DBSPACE to view information about database space when the server is not running.  |
| DSMSERV DISPLAY LOGVOLUMES     | Use DSMSERV DISPLAY LOG to display information about recovery logs including the active log, the mirror for the active log, the failover directory for the archive log, and the overflow location for logs.   |
| DSMSERV DUMPDB                 | The operation that this utility performed is no longer needed.  |

Table 12. Deleted utilities (continued)

| Deleted utility   | Comments   |
|---|--|
| DSMSERV EXTEND LOG  | This utility is replaced by the following server options:<br>ACTIVELOGSIZE<br>ACTIVELOGDIR<br>MIRRORLOGDIR<br><br>With these options, you can add recovery log space if the log is full when the server is down. |
| DSMSERV LOADDB  | The operation that this utility performed is no longer needed.   |
| DSMSERV RESTORE DB<br>Restore a single database volume to its most current state<br>Restore a database to a point in time when a volume history file is unavailable | The server does not track individual database volumes in V6.1.<br><br>The volume history file is required to perform database restore operations.  |
| DSMSERV UNLOADDB  | The operation that this utility performed is no longer needed.   |

## Deleted server options

When you start the server, you might receive warning messages about server options that are not supported, but that do not appear in this list of deleted options. V5 releases tolerated the presence of some server options that were not supported by the server. The V6.1 server flags such options by issuing warning messages. You can ignore the error, or update the server options file and restart the server.

Table 13. Deleted server options

| Deleted option            | Comments   |
|---------------------------|--|
| BUFPOOLSIZE               | The server adjusts the value of buffer pool size dynamically.  |
| DBPAGESHADOW              | The option is no longer needed.  |
| DBPAGESHADOWFILE          | The option is no longer needed.  |
| LOGPOOLSIZE               | The server uses its own fixed-size recovery log buffer pool.   |
| LOGWARNFULLPERCENT        | The option is no longer needed.  |
| MIRRORREAD<br>MIRRORWRITE | Mirroring of the active log is supported, but not of the database. Provide availability protection for the database by locating the database on devices that have high availability characteristics. |
| SELFTUNEBUFPOOLSIZE       | The server adjusts the buffer pool size dynamically.   |

---

## Changes to the **SELECT** command

The **SELECT** command in previous versions of the server allowed syntax that did not always conform to SQL syntax rules. With V6.1, the server conforms to SQL syntax rules in use by its database manager, the DB2 program. Some examples illustrate changes that you might need to make to **SELECT** statements that you use.

“**LIKE** predicate for a nested **SELECT** statement”

“**SELECT** statements for time calculation”

“The **index\_keyseq** and **index\_order** columns”

“Access to database objects using the **SELECT** command”

“Retrieval of information from more than one database table” on page xxxviii

“Results of the **SELECT** command for the **DISK** device class” on page xxxviii

“Extra spaces appearing in output” on page xxxviii

“Data types for arithmetic operations” on page xxxviii

### **LIKE predicate for a nested SELECT statement**

You cannot use the **LIKE** predicate for a nested **SELECT** statement. For example, you receive an error if you use the **LIKE** predicate as in this statement:

```
select * from volumeusage where volume_name like (select distinct volume_name
from volumeusage where node_name='node1')
```

Replace such usage with the **in** parameter, as in this statement:

```
select * from volumeusage where volume_name in (select distinct volume_name
from volumeusage where node_name='node1')
```

### **SELECT statements for time calculation**

Labeled duration cannot be compared in a **SELECT** statement. For example, the following statement results in an SQL error:

```
select * from actlog where (current_time-date_time) seconds <= 60 seconds
```

The following statement is an example of correct usage:

```
select * from actlog where second(current_time-date_time) <= 60
```

### **The index\_keyseq and index\_order columns**

The system catalog tables SYSCAT.COLUMNS and SYSCAT.TABLES are now processed by the database manager, instead of directly by the Tivoli Storage Manager server. The **INDEX\_KEYSEQ** and **INDEX\_ORDER** columns are not available. Use the **KEYSEQ** column instead. For information, use the search string **keyseq** in the following information center: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>.

### **Access to database objects using the SELECT command**

The database objects that can be accessed using the **SELECT** command are the same as for earlier versions of the server, with some additions for new functions in V6.1.

However, the SYSCAT.COLUMNS and SYSCAT.TABLES catalog tables now include all database objects that are known to the server, including some objects that cannot be accessed through the **SELECT** command. You receive an error message if



a SELECT command includes an attempt to access one of these objects.

## Retrieval of information from more than one database table

To retrieve information from more than one table, use a join process. Many types of join processes can be used.

For example, the following command, which worked with earlier versions of the server, no longer works:

```
select entity,activity,sum(bytes),sum(end_time-start_time),sum(affected),sum(failed),sum(mediaw)
from summary where entity in (select node_name from nodes) and
cast((current_timestamp-start_time)hours as decimal)<24 group by entity,activity
```

You can declare names for columns that are retrieved from multiple tables so that a conditional statement can be run with the results that you want from the SELECT command. For example:

```
select entity,activity,sum(bytes),sum(end_time-start_time),sum(affected),sum(failed),sum(mediaw)
from summary su, nodes nd where su.entity=nd.node_name and
cast((current_timestamp-start_time)hours as decimal)<24 group by entity,activity
```

## Results of the SELECT command for the DISK device class

Results when you use the SELECT command to get information from the DEVCLASSES table have changed slightly in V6.1 for the **DISK** device class.

In previous releases, the SHARED field was blank (null) for the **DISK** device class. In V6.1, the SHARED field contains the value N0. The SHARED field does not apply to the **DISK** device class, and the value N0 can be ignored.

## Extra spaces appearing in output

Spaces might appear in output where they did not appear before. If trailing spaces appear in your output, such as in the following **tabschema** output example, you can use the **RTRIM** scalar function to remove them.

```
dsmdmcc -errorlog=errorlog -id=admin -pa=admin -comma -dataonly=y
'select tabschema,tabname from tables'
```

```
SYSCAT ,ATTRIBUTES
SYSCAT ,AUDITPOLICIES
SYSCAT ,AUDITUSE
```

For example, if you are writing scripts for automation and need to strip out the additional spaces, you can use the **RTRIM** scalar function:

```
select rtrim(tabschema) as tabschema, tabname from syscat.tables
```

## Data types for arithmetic operations

Changes in how data types for arithmetic operations are handled might require changes to SELECT commands that worked in earlier versions of the server.

For example, the following command causes an arithmetic overflow error because of the **SUM** statement:

```
select node_name,sum(capacity) as capacity,
sum(capacity * (pct_util/100)) as used from filespaces group by node_name
```

To make the command compatible with V6.1, add the **CAST** function to convert the items in the **SUM** statement to decimal data types:

```
|      select node_name,sum(capacity) as capacity,  
|      sum(cast(capacity as decimal) * cast((pct_util/100) as decimal)) as used from  
|      tablespaces group by node_name
```



---

## Chapter 1. Administering the server from the command line

Tivoli Storage Manager provides several different command line interfaces. The interface you choose depends on the tasks that you want to perform and accessibility.

Tivoli Storage Manager provides three command line interfaces:

- Administrative command-line client
- Server console
- Administration Center command line

The administrative command-line client is a program that runs on a file server, workstation, or mainframe. It is installed as part of the Tivoli Storage Manager server installation process. With the administrative client, you can issue any and all server commands.

Compared to the administrative client, the capabilities of the server console are somewhat limited. For example, you cannot issue certain commands from the server console, and you cannot specify that certain commands process before other commands can be issued. (This procedure can be useful if, for example, you want to run two commands in quick succession.) Furthermore, because the server console is a DOS window on the machine on which the server is installed, you must be physically located at the server machine to use the console. The administrative client, on the other hand, can be accessed remotely from a different location. You cannot route commands to other servers from the server console.

The Administration Center is a Web-based, task-oriented interface for centrally configuring and managing Tivoli Storage Manager servers. The Administration Center provides a command line interface that you can use if necessary. For example, you might want to use the command line interface to perform Tivoli Storage Manager functions that are limited or not supported in the Administration Center. Using the command line in the Administration Center, you can issue any and all server commands.

---

### Issuing commands from the administrative client

The administrative command-line client is a program that runs on a file server, workstation, or mainframe.

Ensure that your administrative client and your server are running in compatible languages. See “LANGUAGE” on page 1204 for language and locale options. If your client and server are using different languages, the messages that Tivoli Storage Manager generates might not be understandable.

## Starting and stopping the administrative client

Use the DSMADMC command to start an administrative client session.

The Tivoli Storage Manager server must be running before an administrative client can connect. For instructions about starting the server, see the *Installation Guide*.

To start an administrative client session in command-line mode, enter this command on your workstation:

```
dsmadmc -id=admin -password=admin -dataonly=yes
```

**Note:** If you do not want to be prompted for a user ID and password, enter the DSMADMC command using the -ID and -PASSWORD options as shown.

Stop an administrative command-line client session by entering the following command on your workstation:

```
quit
```

## Monitoring server activities from the administrative client

To monitor Tivoli Storage Manager activities, such as server migration and client logons, run the administrative client in console mode. You cannot enter any administrative commands in console mode.

To start an administrative client session in console mode, enter:

```
dsmadmc -consolemode
```

You are prompted for a password if authentication is turned on for the server. If you do not want to be prompted for your user ID and password, enter the DSMADMC command with the -ID and -PASSWORD options.

To end an administrative client session in console mode, see Table 14.

*Table 14. Keyboard break sequences*

| Environment                   | Break Sequence     |
|-------------------------------|--------------------|
| UNIX-based and Linux® clients | Ctrl+C             |
| Windows                       | Ctrl+C, Ctrl+Break |
| TSO                           | ATTN               |

## Monitoring removable-media mounts from the administrative client

To monitor the mounting and dismounting of removable media, run the administrative client in mount mode. When the client is running in mount mode, you cannot enter any administrative commands.

To start an administrative client session in mount mode, enter:

```
dsmadmc -mountmode
```

You are prompted for a password if authentication is turned on for the server. If you do not want to be prompted for your user ID and password, enter the DSMADMC command with the -ID and -PASSWORD options.

To end an administrative client session in mount mode, see Table 14.

## Processing commands one at a time from the administrative client

Use batch mode to enter a single administrative command. Your administrative client session automatically ends when the command has processed.

To start an administrative client session in batch mode, enter:

```
dsmdmc server_command
```

If you do not want to be prompted for your user ID and password, you can enter the DSMADMC command with the -ID and -PASSWORD options.

In batch mode, you must enter the complete command on one line. If a command does not fit on one line, enter the command by using a macro or a script. If you specify a parameter with a string of text using batch mode, enclose the text in single quotation marks ( ' ') in the macro. Do not use double quotation marks for commands in batch mode, because your operating system might not parse the quotation marks correctly.

## Processing a series of commands from the administrative client

Use the interactive mode to process 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.
- Commands were not issued from the administrative client session for the length of time specified with the IDLETIMEOUT server option.
- The administrative client session was canceled with the CANCEL SESSION command.

You can use continuation characters when using interactive mode. For more information, see “Entering long commands” on page 11.

To start an administrative session in interactive mode, enter:

```
dsmdmc
```

Do not enter a server command with the DSMADMC command. Doing so will start the administrative client in batch, not interactive, mode. For example, do not enter:

```
dsmdmc server_command
```

You can automatically restart your administrative client session by entering another command each time the **tsm: servername >** prompt appears.

## Formatting output from commands

Tivoli Storage Manager formats the output processed from commands according to your screen or window width.

If the width of your screen or window is not wide enough to display the output horizontally, Tivoli Storage Manager arranges and displays the information vertically.

You can format the output of QUERY commands using the DISPLAYMODE and OUTFILE administrative client options.

## Saving command output to a specified location

The most common use for redirecting output is to save the output from query commands to a specified file or program. You can then browse the contents of the file or, in some cases, print the contents.

Some platforms support redirection of output using special characters such as >, >>, and |. You can save the output from a command by entering redirection characters at the end of the command. Redirection characters direct the output of a command to a file or program you specify instead of to your screen. To redirect output, leave a blank between the redirection character and the file or program name. See the following examples.

**Note:** When redirecting output, follow the naming conventions of the operating system running your administrative client.

| If you want to:  | Enter this:  |
|--|--|
| Redirect the output of a QUERY DOMAIN command to a new file in batch or interactive mode                 | <pre>dsmadmc -id=xxx -pa=xxx query domain acctg dominfo.acc</pre> <p>A single greater-than sign (&gt;) indicates that Tivoli Storage Manager redirects the output to a new file or writes over an existing file.</p>             |
| Append the output of a QUERY DOMAIN command to the end of an existing file in batch or interactive mode  | <pre>dsmadmc -id=xxx -pa=xxx query domain acctg &gt;&gt; dominfo.acc</pre> <p>Double greater-than signs (&gt;&gt;) indicates that Tivoli Storage Manager appends the output to the end of an existing file.</p>                  |
| Redirect all output from an administrative client session in console mode to a program called filter.exe | <pre>dsmadmc -console -id=admin -password=xxx   filter.exe</pre> <p>The program can be set up to monitor the output for individual messages as they occur and take appropriate action, such as sending mail to another user.</p> |

You can also redirect all output to a specified file by specifying the -OUTFILE option with a destination file name. For example, enter:

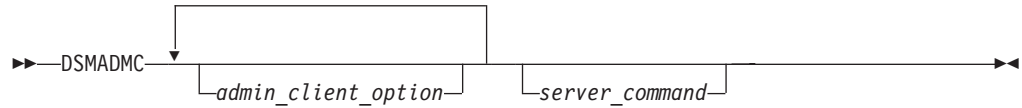
```
dsmadmc -id=sullivan -password=secret -consolemode -outfile=save.out
```



## Administrative client options

In all administrative client modes, you can use options to modify administrative client session responses.

### DSMADMC



Here is an example of a task and how to use the administrative client options: You can enter the DSMADMC command with your user ID and password by using the `-ID` and `-PASSWORD` options, respectively if you do not want to be prompted for that information. To have Tivoli Storage Manager redirect all output to a file, specify the `-OUTFILE` option with a destination file name. For example, to issue the `QUERY NODE` command in batch mode with the output redirected to the `SAVE.OUT` file, enter:

```
dsmadmc -id=sullivan -password=secret -outfile=save.out query node
```

### Options

Administrative client options can be specified with the DSMADMC command and are valid from an administrative client session only. You can type an option in uppercase letters, lowercase letters, or any combination. Uppercase letters denote the shortest acceptable abbreviation. If an option appears entirely in uppercase letters, you cannot abbreviate it.

#### **-ALWAYSPrompt**

Specifies that a command prompt is displayed if the input is from the keyboard or if it is redirected (for example, from a file). If this option is not specified and the input is redirected, the command prompt is not written.

If the input is redirected, only the command output is displayed. If this option is specified, the command prompt and the command output are displayed.

#### **-CHECKAliasalt**

Allows the administrative client to recognize an alias for the `HALT` command as set in the `ALIASHALT` server option. See “`ALIASHALT`” on page 1184 for details.

#### **-COMMA delimited**

Specifies that any tabular output from a server query is to be formatted as comma-separated strings rather than in readable format. This option is intended to be used primarily when redirecting the output of an SQL query (`SELECT` command). The comma-separated value format is a standard data format, which can be processed by many common programs, including spreadsheets, databases, and report generators.

#### **-CONsolemode**

Specifies that Tivoli Storage Manager runs in console mode. All server console output is echoed to your screen, excluding items such as responses to query commands issued from the console, trace output, or any system messages that might appear on the console.

**-DATAONLY=*List or Table***

Specifies whether product version information and output headers display with the output. The default is NO.

**NO**

Specifies that the product version information and output column headers display.

**YES**

Suppresses the product version information and output column headers.

**-DISPLaymode=*List or Table***

Allows you to force the QUERY output to tabular or list format regardless of the command line window column width.

If you are using the -DISPLAYMODE option and you want the output to go to a file, do not specify the -OUTFILE option. Use redirection to write to the file.

**-ID=*userid***

Specifies the administrator's user ID.

**-Itemcommit**

Specifies that Tivoli Storage Manager commits commands inside a script or a macro as each command is processed.

**-MOUNTmode**

Specifies that Tivoli Storage Manager runs in mount mode. All server removable-media mount messages are echoed to your screen.

**-NEWLINEAFTERPrompt**

Specifies that a newline character is written immediately after the command prompt and commands entered from the keyboard appear immediately below the prompt. If this option is not specified, commands entered from the keyboard appear immediately to the right of the prompt.

**-NOConfirm**

Specifies that you do not want Tivoli Storage Manager to request confirmation before processing commands that affect the availability of the server or data managed by the server.

**-OUTfile**

Specifies that output from a server query is formatted one line per query. This option is available in batch mode only.

**-OUTfile=*filename***

Specifies that output from a server query is redirected to a specified file. In batch mode, output is redirected to a file you specify and the format of the output matches the format of the output on your screen.

In interactive, console, or mount mode sessions, output displays on your screen.

**-Password=*password***

Specifies the administrator's password.

**-Quiet**

Specifies that Tivoli Storage Manager does not display standard output messages to your screen. However, when you use this option, certain error messages still appear.

**-Serveraddress**

Specifies the server stanza in the dsm.sys file. The client uses the server stanza

to determine the server it will connect to. The SERVERADDRESS option is supported by administrative clients running on UNIX, Linux, and Macintosh operating systems only.

**-TABdelimited**

Specifies that any tabular output from a server query is to be formatted as tab-separated strings rather than in readable format. This option is intended to be used primarily when redirecting the output of an SQL query (SELECT command). The tab-separated value format is a standard data format, which can be processed by many common programs, including spreadsheets, databases, and report generators.

**-TCPPort**

Specifies a TCP/IP port address for a Tivoli Storage Manager server. The TCPPORT option is only supported by administrative clients running on Windows operating systems and is valid on the Windows administrative client command line.

**-TCPServeraddress**

Specifies a TCP/IP server address for a Tivoli Storage Manager server. The TCPSERVERADDRESS option is only supported by administrative clients running on Windows operating systems and is valid on the Windows administrative client command line.

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.

---

## Issuing commands from the Administration Center

A command-line interface is available from all of the main server tables in the Administration Center. To access the command line, select a server, click **Select Action**, and select **Use Command Line**.

---

## Issuing commands from the server console

Tivoli Storage Manager provides a user ID named SERVER\_CONSOLE that allows you to issue commands and administer the server from the server console after Tivoli Storage Manager is installed. At installation, SERVER\_CONSOLE is automatically registered as an administrator and is given system authority.

If you have system privilege, you can revoke or grant new privileges to the SERVER\_CONSOLE user ID. However, you cannot:

- Register or update the SERVER\_CONSOLE user ID
- Lock or unlock the SERVER\_CONSOLE user ID
- Rename the SERVER\_CONSOLE user ID
- Remove SERVER\_CONSOLE user ID
- Route commands from the SERVER\_CONSOLE user ID

**Note:** Not all Tivoli Storage Manager commands are supported by the server console. You cannot specify the WAIT parameter from the server console.

---

## Entering administrative commands

Commands consist of command names and usually parameters and variables. Syntax diagrams depict the rules to follow when entering commands.

### Reading syntax diagrams

To read a syntax diagram for entering a command, follow the path of the line. Read from left to right and from 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.

### Command names

The command name can consist of a single action word, such as **HALT**, or it can consist of an action word and an object for the action, such as **DEFINE DOMAIN**. You can enter the command in any column of the input line.

Enter the entire command name or the abbreviation that is specified in the syntax diagram for the command. Uppercase letters denote the shortest acceptable abbreviation. If a command appears entirely in uppercase letters, you cannot abbreviate it. You can enter the command in uppercase letters, lowercase letters, or any combination. In this example, you can enter **CMDNA**, **CMDNAM**, or **CMDNAME** in any combination of uppercase and lowercase letters.

▶—CMDName—▶◀

**Note:** Command names in descriptive text are always capitalized.

### Required parameters

When a parameter is on the same line as the command name, the parameter is required. When two or more parameter values are in a stack and one of them is on the line, you *must* specify one value.

In this example, you must enter **PARMNAME=A**, **PARMNAME=B**, or **PARMNAME=C**. Do not include any blanks immediately before or after the equal sign (=).

▶—PARMName==

|   |
|---|
| A |
| B |
| C |

—▶◀

### Optional parameters

When a parameter is below the line, the parameter is optional. In this example, you can enter **PARMNAME=A** or nothing at all. Do not include any blanks immediately before or after the equal sign (=).



When two or more parameter values are in a stack below the line, all of them are optional. In this example, you can enter PARMNAME=A, PARMNAME=B, PARMNAME=C, or nothing at all. Do not include any blanks immediately before or after the equal sign (=).



## Defaults

Defaults are above the line. The system uses the default unless you override it. You can override the default by entering an option from the stack below the line.

In this example, PARMNAME=A is the default. You can also enter PARMNAME=A, PARMNAME=B, or PARMNAME=C. Do not include any blanks before or after the equal sign (=).



## Variables

Highlighted lowercase items (*like this*) denote variables. In these examples, *var\_name* represents variables::



## Special characters

You must code these symbols exactly as they appear in the syntax diagram.

- \* Asterisk
- :
- ,
- = Equal sign
- Hyphen
- ( ) Parentheses
- .

## Repeating values

An arrow returning to the left means that the item can be repeated. A character within the arrow means that you must separate repeated items with that character.



## Repeatable choices

A stack of values followed by an arrow returning to the left means that you can select more than one value or, when permitted, repeat a single item. 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 (=).



## Footnotes

Footnotes are enclosed in parentheses.

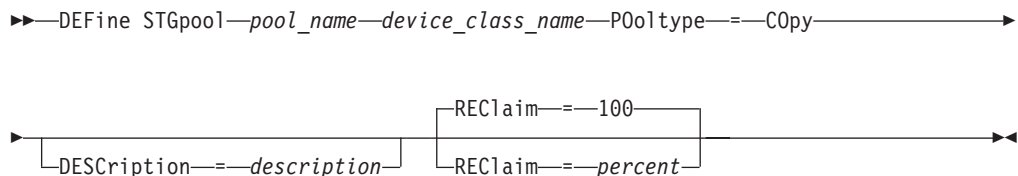


### Notes:

- 1 You can specify up to five file names.

## Entering parameters

The order in which you enter parameters can be important. The following example shows a portion of the command for defining a copy storage pool:



The first two parameters in this command (`pool_name` and `device_class_name`) are required parameters. `pool_name` and `device_class_name` are also positional. That is, they must be entered in the order shown, immediately after the command name. The **POOLTYPE** parameter is a required keyword parameter. **DESCRIPTION** and **RECLAIM** are optional keyword parameters. Keyword parameters are identified by an equal sign that specifies a specific value or a variable. Keyword parameters must follow any positional parameters in a command.

The following command entries, in which the keyword parameters are ordered differently, are both acceptable:

```
define stgpool mycopypool mydeviceclass pooltype=copy description=engineering  
reclaim=50  
define stgpool mycopypool mydeviceclass description=engineering pooltype=copy  
reclaim=50
```

The following example, in which one of the positional parameters follows a keyword parameter, is not acceptable:

```
define stgpool mycopypool pooltype=copy mydeviceclass description=engineering  
reclaim=50
```

## Syntax fragments

Some diagrams, because of their length, must display parts of the syntax with fragments. 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. A heading with the fragment name identifies the expanded fragment. Commands appearing directly on the line are required.

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

►► | Fragment | ◄◄

### Fragment:



## Entering long commands

Continuation characters are useful when you want to process a command that is longer than your screen or window width. You can use continuation characters in the interactive mode of the administrative client.

Without continuation characters you can enter up to 256 characters. With continuation characters you can enter up to 1500 characters.

**Note:** In the MACRO command, the maximums apply *after* any substitution variables have been applied.

With continuation characters, you can do the following:

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

For example:

```
register admin pease mypasswd -  
contact="david, ext1234"
```

- Continue a list of values by entering 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
```

- Continue a string of values enclosed in quotation marks by entering 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"
```

Tivoli Storage Manager 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.

## Naming Tivoli Storage Manager objects

Tivoli Storage Manager restricts the number and type of characters that you can use to name objects.

The following characters are available for defining object names.

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

The following table shows the maximum length of characters permitted for naming objects.

| Type of Name   | Maximum Length |
|--|----------------|
| Administrators, client option sets, client nodes, passwords, server groups, server, names, virtual file space names            | 64             |
| Restartable export identifiers   | 64             |
| High-level and low-level TCP/IP (IPv4 or IPv6) addresses   | 64             |
| Device classes, drives, libraries, management classes, policy domains, profiles, schedules scripts, backup sets, storage pools | 30             |

When you use DEFINE commands to define database, recovery log, and storage pool volumes, the naming convention for the volume name is dependent on the type of sequential access media or random access media you are using. Refer to the specific VOLUME command for details.

## Using wildcard characters to specify object names

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

*Table 15. Wildcard characters by operating system*

| Operating system                               | Match any | Match exactly one |
|--|-----------|-------------------|
| AIX, HP-UX, Linux, OS/2®, Sun Solaris, Windows | *         | ?                 |
| TSO  | *         | %                 |

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*
```

If your system uses a question mark as the *match-exactly-one* character, and you want to query the management classes in POLICYSET1 in DOMAIN1, you can enter:

```
query mgmtclass domain1 policyset1 mc?
```

Tivoli Storage Manager displays information about management classes with names MC.

Table 16 shows additional examples of using wildcard characters to match any characters.

*Table 16. Match-any character*

| Pattern  | Matches             | Does not match    |
|----------|---------------------|-------------------|
| ab*      | ab, abb, abxxx      | a, b, aa, bb      |
| ab*rs    | abrs, abtrs, abrsrs | ars, aabrs, abrss |
| ab*ef*rs | abefrs, abefghrs    | abefr, abers      |

Table 17 shows additional examples of 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 17. Match-exactly-one character*

| Pattern  | Matches  | Does not match           |
|----------|----------|--------------------------|
| ab?      | abc      | ab, abab, abzzzz         |
| ab?rs    | abrs     | abrs, abllrs             |
| ab?ef?rs | abdefjrs | abefrs, abdefrs, abefjrs |

Table 17. Match-exactly-one character (continued)

| Pattern | Matches        | Does not match      |
|---------|----------------|---------------------|
| ab??rs  | abcdrs, abzzrs | abrs, abjrs, abkkrs |

## Specifying descriptions in keyword parameters

If a description (a string of text) for a parameter begins with a single or double quotation mark, or contains any embedded blanks or equal signs, you must surround the value with 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.

For example, to register a new client node named Louie, with a password of secret, and with his title included as contact information, enter:

```
register node louie secret contact="manager of dept. 61f"
```

The following table presents ways of entering a description for the CONTACT parameter. The value can contain quotation marks, embedded blanks, or equal signs.

| For this description         | Enter this  |
|------------------------------|---|
| manager                      | contact=manager   |
| manager's                    | contact="manager's" or contact='manager's'  |
| "manager"                    | contact="\"manager\"" or contact='\"manager\"'                                      |
| manager's report             | contact="manager's report" or contact='manager's report'                            |
| manager's "report"           | contact='manager's "report"'  |
| manager=dept. 61f            | contact='manager=dept. 61f'   |
| manager reports to dept. 61f | contact='manager reports to dept. 61f' or<br>contact="manager reports to dept. 61f" |

## Controlling command processing

You can run some Tivoli Storage Manager commands sequentially or concurrently with other commands. You can also route commands from one server to other servers for processing.

### Server command processing

Tivoli Storage Manager processes administrator commands either in the foreground or in the background. Commands that process in the foreground must complete before you can issue another command. When commands are processing in the background, you can issue additional commands at any time.

Most Tivoli Storage Manager commands process in the foreground. For some commands that normally process in the background (for example, BACKUP DB), you can specify the WAIT parameter (WAIT=YES) with the command so that the command processes in the foreground. You might want to process a command in the foreground rather than in the background for any of the following reasons:

- To quickly determine whether a command completed successfully. When you issue a command that processes in the foreground, Tivoli Storage Manager sends a confirmation message indicating that the command completed successfully. If you process the command in the background, you need to open operational reporting or query the activity log to determine whether the command completed successfully.
- To monitor server activities (for example, messages) on the administrative client as a command is being processed. This might be preferable to searching a long activity log after the command has completed.
- To be able to start another process immediately after a command has completed. For example, you might specify WAIT=YES for a command that takes a short time to process so that, when the processing completes, you can immediately start processing another command.
- To serialize commands in an administrative script when it is important that one command completes before another begins.

Check the individual command description to determine whether a command has a WAIT parameter.

You can cancel commands that are processed in the foreground from the server console or from another administrative client session.

Each background process is assigned a process number. Use the QUERY PROCESS command to obtain the status and process number of a background process.

**Note:**

- If you are defining a schedule with a command that specifies WAIT=NO (the default), and you issue QUERY EVENT to determine the status of your scheduled operation, failed operations will report an event status of COMPLETED with a return of OK. In order for the QUERY EVENT output to reflect the failed status, the WAIT parameter must be set to YES. This will run the scheduled operation in the foreground and inform you of the status when it completes.
- You cannot process commands in the foreground from the server console.

## Cancelling commands

Use the CANCEL PROCESS command to cancel commands that generate background processes.

Use the QUERY PROCESS command to obtain the status and process number of 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.

When you issue a QUERY command from the administrative client, multiple screens of output might be generated. If this occurs and additional output is not needed, you can cancel the display of output to the client workstation. Doing so does not end the processing of the command.

---

## Performing tasks concurrently on multiple servers

Command routing allows you to route commands to one or more servers for processing and then collect the output from these servers.

To route commands to other servers, you must have the same administrator ID and password as well as the required administrative authority on each server to which the command is being routed. You cannot route commands to other servers from the server console.

After the command has completed processing on all servers, the output displays, in its entirety, for each server. For example, the output from SERVER\_A displays in its entirety, followed by the output from SERVER\_B. The output includes summary messages for each individual server and identifies which server processed the output. Return codes indicate whether commands processed on the servers successfully. These return codes include one of three severities: 0, ERROR, or WARNING.

Each server that is identified as the target of a routed command must first be defined using the DEFINE SERVER command. The command is automatically routed to all servers specified as members of a server group or to individual servers specified with the command. For details about setting up and managing multiple servers for command routing, see the *Administrator's Guide*.

The following examples describe how to route the QUERY STGPOOL command to one server, multiple servers, a server group, multiple server groups, or a combination of servers and server groups. Each server or server group in a list must be separated with a comma, without spaces.

### Routing commands to a single server

To route the QUERY STGPOOL command to a server named ASTRO, enter:

```
astro: query stgpool
```

The colon after the server name indicates the end of the routing information. This is also called the *server prefix*. Another way to indicate the end of routing information is to use parentheses around the server name, for example:

```
(astro) query stgpool
```

### Routing commands to multiple servers

To route the QUERY STGPOOL command to multiple servers named HD\_QTR, MIDAS, SATURN, enter:

```
hd_qtr,midas,saturn: query stgpool
```

If the first server has not been defined to Tivoli Storage Manager, the command is routed to the next defined server in the list of servers.

You can also enter the command this way:

```
(hd_qtr,midas,saturn) query stgpool
```

## Routing commands to a server group

In this example, the server group ADMIN has servers named SECURITY, PAYROLL, PERSONNEL defined as group members. The command is routed to each of these servers.

To route the QUERY STGPOOL command to the server group named ADMIN, enter:

```
admin: query stgpool
```

You can also enter the command this way:

```
(admin) query stgpool
```

## Routing commands to server groups

In this example, the server group ADMIN2 has servers SERVER\_A, SERVER\_B, and SERVER\_C defined as group members, and server group ADMIN3 has servers ASTRO, GUMBY, and CRUSTY defined as group members. The command is routed to servers SERVER\_A, SERVER\_B, SERVER\_C, ASTRO, GUMBY, and CRUSTY.

To route the QUERY STGPOOL command to two server groups that are named ADMIN2 and ADMIN3, enter:

```
admin2,admin3: query stgpool
```

You can also enter the command this way:

```
(admin2,admin3) query stgpool
```

## Routing commands to two servers and a server group

In this example, the server group DEV\_GROUP has servers SALES, MARKETING, and STAFF defined as group members. The command is routed to servers SALES, MARKETING, STAFF, MERCURY, and JUPITER.

To route the QUERY STGPOOL command to a server group named DEV\_GROUP and to the servers named MERCURY and JUPITER, enter:

```
dev_group,mercury,jupiter: query stgpool
```

You can also enter the command this way:

```
(dev_group,mercury,jupiter) query stgpool
```

## Routing commands inside scripts

When routing commands inside scripts, you must enclose the server or server group in parentheses and omit the colon. Otherwise, the command will not be routed when the RUN command is issued, and will only be run on the server where the RUN command is issued.

For example, to route the QUERY STGPOOL command inside a script:

1. Define a script called QU\_STG to route it to the DEV\_GROUP server group.  

```
define script qu_stg "(dev_group) query stgpool"
```
2. Run the QU\_STG script:  

```
run qu_stg
```

In this example, the server group DEV\_GROUP has servers SALES, MARKETING, and STAFF defined as group members. The QUERY STGPOOL command is routed to these servers.

---

## Privilege classes for commands

The authority granted to an administrator through the privilege class determines which administrative commands that the administrator can issue.

There are five administrator privilege classes in Tivoli Storage Manager:

- System
- Policy
- Storage
- Operator

After an administrator has been registered using the REGISTER ADMIN command, the administrator can issue a limited set of commands, including all query commands. When you install Tivoli Storage Manager, the server console is defined as a system administrator named SERVER\_CONSOLE and is granted system privilege.

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 Tivoli Storage Manager. With system privilege, an administrator can issue any administrative command and has authority to manage all policy domains and all storage pools.

Table 18 on page 19 lists the commands that administrators with system privilege can issue. In some cases administrators with lower levels of authority, for example, unrestricted storage privilege, can also issue these commands. In addition, the REQSYSAUTHOUTFILE server option can be used to specify that certain commands require system privilege if they cause Tivoli Storage Manager to write to an external file. For more information about this server option see “REQSYSAUTHOUTFILE” on page 1223.



Table 18. System privilege commands

| Command Name                 | Command Name                 |
|------------------------------|------------------------------|
| AUDIT LICENSES               | DEFINE STGPOOL               |
| ACCEPT DATE                  | DEFINE SUBSCRIPTION          |
| BEGIN EVENTLOGGING           | DEFINE VIRTUALFSMAPPING      |
| CANCEL EXPIRATION            | DEFINE VOLUME                |
| CANCEL PROCESS               | DELETE BACKUPSET             |
| CANCEL REQUEST               | DELETE CLIENTOPT             |
| CANCEL RESTORE               | DELETE CLOPTSET              |
| CLEAN DRIVE                  | DEFINE COLLOGGROUP           |
| COPY ACTIVATEDATA            | DEFINE COLLOCMEMBER          |
| COPY DOMAIN                  | DELETE DOMAIN                |
| COPY POLICYSET               | DELETE DRIVE                 |
| COPY PROFILE                 | DELETE EVENTSERVER           |
| COPY SCHEDULE (See note.)    | DELETE GRPMEMBER             |
| COPY SCRIPT                  | DELETE LIBRARY               |
| COPY SERVERGROUP             | DELETE MACHINE               |
| DEFINE BACKUPSET             | DELETE MACHNODEASSOCIATION   |
| DEFINE CLIENTACTION          | DELETE NODEGROUP             |
| DEFINE CLIENTOPT             | DELETE NODEGROUPMEMBER       |
| DEFINE CLOPTSET              | DELETE PROFASSOCIATION       |
| DEFINE COLLOGGROUP           | DELETE PROFILE               |
| DEFINE COLLOCMEMBER          | DELETE RECMEDMACHASSOCIATION |
| DEFINE DEVCLASS              | DELETE RECOVERYMEDIA         |
| DEFINE DOMAIN                | DELETE SCHEDULE (See note.)  |
| DEFINE DRIVE                 | DELETE SCRIPT                |
| DEFINE EVENTSERVER           | DELETE SERVER                |
| DEFINE GRPMEMBER             | DELETE SERVERGROUP           |
| DEFINE LIBRARY               | DELETE SPACETRIGGER          |
| DEFINE MACHINE               | DELETE STGPOOL               |
| DEFINE MACHNODEASSOCIATION   | DELETE SUBSCRIBER            |
| DEFINE NODEGROUP             | DELETE SUBSCRIPTION          |
| DEFINE NODEGROUPMEMBER       | DELETE VIRTUALFSMAPPING      |
| DEFINE PATH                  | DISABLE EVENTS               |
| DEFINE PROFASSOCIATION       | ENABLE EVENTS                |
| DEFINE PROFILE               | END EVENTLOGGING             |
| DEFINE RECMEDMACHASSOCIATION | EXPIRE INVENTORY             |
| DEFINE RECOVERYMEDIA         | EXPORT ADMIN                 |
| DEFINE SCHEDULE (See note.)  | EXPORT NODE                  |
| DEFINE SCRIPT                | EXPORT POLICY                |
| DEFINE SERVER                | EXPORT SERVER                |
| DEFINE SERVERGROUP           | GENERATE BACKUPSET           |
| DEFINE SPACETRIGGER          | GRANT AUTHORITY              |

Table 18. System privilege commands (continued)

| Command Name                   | Command Name              |
|--------------------------------|---------------------------|
| GRANT PROXYNODE                | SET CROSSDEFINE           |
| IDENTIFY DUPLICATES            | SET DBRECOVERY            |
| IMPORT NODE                    | SET DRMACTIVEDATASTGPOOL  |
| IMPORT POLICY                  | SET DRMCHECKLABEL         |
| IMPORT SERVER                  | SET DRMCMDFILENAME        |
| INSERT MACHINE                 | SET DRMCOPYSTGPOOL        |
| LABEL LIBVOLUME                | SET DRMCOURIERNAME        |
| LOCK ADMIN                     | SET DRMDBBACKUPEXPIREDAYS |
| LOCK PROFILE                   | SET DRMFILEPROCESS        |
| MIGRATE STGPOOL                | SET DRMINSTRPREFIX        |
| MOVE DRMEDIA                   | SET DRMNOTMOUNTABLENAME   |
| MOVE MEDIA                     | SET DRMPPLANPREFIX        |
| MOVE GRPMEMBER                 | SET DRMPPLANVPOSTFIX      |
| NOTIFY SUBSCRIBERS             | SET DRMPRIMSTGPOOL        |
| PING SERVER                    | SET DRMRPFEXPIREDAYS      |
| PREPARE                        | SET DRMVAULTNAME          |
| QUERY BACKUPSETCONTENTS        | SET EVENTRETENTION        |
| QUERY MEDIA                    | SET INVALIDPWLIMIT        |
| QUERY RPFCONTENT               | SET LICENSEAUDITPERIOD    |
| QUERY TOC                      | SET MAXCMDRETRIES         |
| RECLAIM STGPOOL                | SET MAXSCHEDSESSIONS      |
| RECONCILE VOLUMES              | SET MINPWLENGTH           |
| REGISTER ADMIN                 | SET PASSEXP               |
| REGISTER LICENSE               | SET QUERYSCHEDPERIOD      |
| REMOVE ADMIN                   | SET RANDOMIZE             |
| RENAME ADMIN                   | SET REGISTRATION          |
| RENAME SCRIPT                  | SET RETRYPERIOD           |
| RENAME SERVERGROUP             | SET SCHEDMODES            |
| RENAME STGPOOL                 | SET SERVERHLADDRESS       |
| RESET PASSEXP                  | SET SERVERLLADDRESS       |
| RESTORE NODE                   | SET SERVERNAME            |
| REVOKE AUTHORITY               | SET SERVERPASSWORD        |
| REVOKE PROXYNODE               | SET SUBFILE               |
| RUN                            | SET TOCLOADRETENTION      |
| SET ACCOUNTING                 | SETOPT                    |
| SET ACTLOGRETENTION            | UNLOCK ADMIN              |
| SET ARCHIVERETENTIONPROTECTION | UNLOCK PROFILE            |
| SET AUTHENTICATION             | UPDATE ADMIN              |
| SET CLIENTACTDURATION          | UPDATE BACKUPSET          |
| SET CONFIGMANAGER              | UPDATE CLIENTOPT          |
| SET CONFIGREFRESH              | UPDATE CLOPTSET           |
| SET CONTEXTMESSAGING           | UPDATE COLLOCGROUP        |

Table 18. System privilege commands (continued)

| Command Name         | Command Name                |
|----------------------|-----------------------------|
| UPDATE DEVCLASS      | UPDATE SCHEDULE (See note.) |
| UPDATE DRIVE         | UPDATE SCRIPT               |
| UPDATE LIBRARY       | UPDATE SERVER               |
| UPDATE LIBVOLUME     | UPDATE SERVERGROUP          |
| UPDATE MACHINE       | UPDATE SPACETRIGGER         |
| UPDATE NODEGROUP     | UPDATE VIRTUALFSMAPPING     |
| UPDATE PATH          | UPDATE VOLHISTORY           |
| UPDATE PROFILE       | VALIDATE LANFREE            |
| UPDATE RECOVERYMEDIA |                             |

**Note:** This command is restricted by the authority granted to an administrator. System privilege is required only for administrative command schedules. System or 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 privilege can be unrestricted, or can be restricted to specific policy domains.

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 19 on page 22 lists the commands that an administrator with policy privilege can issue.

Table 19. Policy privilege commands

| Command Name                | Command Name                  |
|-----------------------------|-------------------------------|
| ACTIVATE POLICYSET          | DELETE POLICYSET              |
| ASSIGN DEFMGMTCLASS         | DELETE PATH                   |
| CLEAN DRIVE                 | DELETE SCHEDULE (See note 2.) |
| BACKUP NODE                 | GENERATE BACKUPSET            |
| COPY MGMTCLASS              | LOCK NODE                     |
| COPY POLICYSET              | QUERY BACKUPSETCONTENTS       |
| COPY SCHEDULE (See note 2.) | REGISTER NODE                 |
| DEFINE ASSOCIATION          | REMOVE NODE                   |
| DEFINE BACKUPSET            | RENAME FILESPACE              |
| DEFINE COPYGROUP            | RENAME NODE                   |
| DEFINE CLIENTACTION         | SET SUMMARYRETENTION          |
| DEFINE CLIENTOPT            | RESTORE NODE                  |
| DEFINE MGMTCLASS            | QUERY TOC                     |
| DEFINE NODEGROUP            | UNLOCK NODE                   |
| DEFINE NODEGROUPMEMBER      | UPDATE BACKUPSET              |
| DEFINE POLICYSET            | UPDATE COPYGROUP              |
| DEFINE SCHEDULE             | UPDATE DOMAIN                 |
| DELETE ASSOCIATION          | UPDATE MGMTCLASS              |
| DELETE BACKUPSET            | UPDATE NODE                   |
| DELETE COPYGROUP            | UPDATE NODEGROUP              |
| DELETE EVENT (See note 1.)  | UPDATE POLICYSET              |
| DELETE FILESPACE            | UPDATE SCHEDULE (See note 2.) |
| DELETE MGMTCLASS            | VALIDATE POLICYSET            |
| DELETE NODEGROUP            |                               |
| DELETE NODEGROUPMEMBER      |                               |

**Notes:**

1. This command can 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. This command is restricted by the authority granted to an administrator. System privilege is required only for administrative command schedules. System or 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 storage privilege can be unrestricted, or can be restricted to specific storage pools.

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 20 lists the commands an administrator with storage privilege can issue.

*Table 20. Storage privilege commands*

| Command Name                  | Command Name               |
|-------------------------------|----------------------------|
| AUDIT LIBRARY                 | DELETE SPACETRIGGER        |
| AUDIT VOLUME (See note.)      | DELETE VIRTUALFSMAPPING    |
| BACKUP DB                     | DELETE VOLHISTORY          |
| BACKUP DEVCONFIG              | DELETE VOLUME (See note.)  |
| BACKUP STGPOOL                | GRANT PROXYNODE            |
| BACKUP VOLHISTORY             | LABEL LIBVOLUME            |
| CHECKIN LIBVOLUME             | MIGRATE STGPOOL            |
| CHECKOUT LIBVOLUME            | MOVE DATA (See note.)      |
| COPY ACTIVATEDATA (See note.) | MOVE MEDIA                 |
| DEFINE COLLOGROUP             | QUERY TAPEALERTMSG         |
| DEFINE COLLOCMEMBER           | RECLAIM STGPOOL            |
| DEFINE DATAMOVER              | RESTORE STGPOOL            |
| DEFINE DEVCLASS               | RESTORE VOLUME             |
| DEFINE DRIVE                  | REVOKE PROXYNODE           |
| DEFINE LIBRARY                | SET TAPEALERTMSG           |
| DEFINE PATH                   | UPDATE COLLOGROUP          |
| DEFINE VIRTUALFSMAPPING       | UPDATE DATAMOVER           |
| DEFINE VOLUME (See note.)     | UPDATE DEVCLASS            |
| DEFINE SPACETRIGGER           | UPDATE DRIVE               |
| DELETE COLLOGROUP             | UPDATE LIBRARY             |
| DELETE COLLOCMEMBER           | UPDATE PATH                |
| DELETE DATAMOVER              | UPDATE SPACETRIGGER        |
| DELETE DEVCLASS               | UPDATE STGPOOL (See note.) |
| DELETE DRIVE                  | UPDATE VIRTUALFSMAPPING    |
| DELETE LIBRARY                |                            |
| DELETE PATH                   |                            |

**Note:** 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 21 on page 24 lists the commands an administrator with operator privilege can issue.

*Table 21. Operator privilege commands*

| Command Name     | Command Name  |
|------------------|---------------|
| CANCEL SESSION   | MOVE DRMEDIA  |
| DISABLE SESSIONS | MOVE MEDIA    |
| DISMOUNT VOLUME  | QUERY MEDIA   |
| ENABLE SESSIONS  | REPLY         |
| HALT             | UPDATE VOLUME |
|                  | VARY          |

## **Commands any administrator can issue**

A limited number of commands can be used by any administrator, even if that administrator has not been granted any specific administrator privileges.

Table 22 lists the commands any registered administrator can issue.

*Table 22. Commands issued by all administrators*

| Command Name        | Command Name           |
|---------------------|------------------------|
| COMMIT              | QUERY NODE             |
| HELP                | QUERY NODEDATA         |
| ISSUE MESSAGE       | QUERY NODEGROUP        |
| MACRO               | QUERY OCCUPANCY        |
| PARALLEL            | QUERY OPTION           |
| QUERY ACTLOG        | QUERY PATH             |
| QUERY ADMIN         | QUERY POLICYSET        |
| QUERY ASSOCIATION   | QUERY PROCESS          |
| QUERY AUDITOCUPANCY | QUERY PROFILE          |
| QUERY BACKUPSET     | QUERY PROXYNODE        |
| QUERY CLOPTSET      | QUERY RECOVERYMEDIA    |
| QUERY COLLOGGROUP   | QUERY REQUEST          |
| QUERY CONTENT       | QUERY RESTORE          |
| QUERY COPYGROUP     | QUERY RPFILE           |
| QUERY DATAMOVER     | QUERY SCHEDULE         |
| QUERY DB            | QUERY SCRIPT           |
| QUERY DBSPACE       | QUERY SERVER           |
| QUERY DEVCLASS      | QUERY SERVERGROUP      |
| QUERY DIRSPACE      | QUERY SESSION          |
| QUERY DOMAIN        | QUERY SPACETRIGGER     |
| QUERY DRIVE         | QUERY STATUS           |
| QUERY DRMEDIA       | QUERY STGPOOL          |
| QUERY DRMSTATUS     | QUERY SUBSCRIBER       |
| QUERY ENABLED       | QUERY SUBSCRIPTION     |
| QUERY EVENT         | QUERY SYSTEM           |
| QUERY EVENTRULES    | QUERY VIRTUALFSMAPPING |
| QUERY EVENTSERVER   | QUERY VOLHISTORY       |
| QUERY FILESPACE     | QUERY VOLUME           |
| QUERY LIBRARY       | QUIT                   |
| QUERY LIBVOLUME     | ROLLBACK               |
| QUERY LICENSE       | SERIAL                 |
| QUERY LOG           | SELECT                 |
| QUERY MACHINE       |                        |
| QUERY MGMTCLASS     |                        |
| QUERY MOUNT         |                        |
| QUERY NASBACKUP     |                        |





---

## Chapter 2. Administrative commands

Administrative commands are available to manage and configure the server.

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
- Examples of using the command
- A list of related commands

## ACCEPT DATE (Accepts the current system date)

Use this command to allow the server to begin normal processing, when the server does not start normal processing because of a discrepancy between the server date and the current date on the system.

When the server does not start normal processing because of a discrepancy between the server date and the current date, this command forces the server to accept the current date and time as valid. If the system time is valid and the server has not been run for an extended time, this command should be run to allow the server to begin normal processing.

**Attention:** If the system date is invalid or the server was created or run previously with an invalid system date and this command is issued, any server processing or command that uses dates can have unexpected results. File expiration can be affected, for example. When the server is started with the correct date, files backed up with future dates will not be considered for expiration until that future date is reached. Files backed up with dates that have passed will expire faster. When the server processing encounters a future date, an error message is issued. See the *Administrator's Guide* for more details.

If the server detects an invalid date or time, server sessions become disabled (as if the DISABLE SESSIONS command had been issued). Expiration, migration, reclamation, and volume history deletion operations are not able to continue processing.

Use the ENABLE SESSIONS ALL command after you issue the ACCEPT DATE command to re-enable sessions to start.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Accept Date—◄◄

### Parameters

None.

### Example: Accept the current system date

Allow the server to accept the current date as the valid date.

```
accept date
```

### Related commands

Table 23. Command related to ACCEPT DATE

| Command         | Description   |
|-----------------|---|
| ENABLE SESSIONS | Resumes server activity following the DISABLE command or the ACCEPT DATE command. |

## ACTIVATE POLICYSET (Activate a new policy set)

Use this command to copy the contents of a policy set to the ACTIVE policy set for the domain. The server uses the rules in the ACTIVE policy set to manage client operations in the domain. You can define multiple policy sets for a policy domain, but only one policy set can be active. The current ACTIVE policy set is replaced by the one you specify when you issue this command. You can modify the ACTIVE policy set only by activating another policy set.

Before activating a policy set, check that the policy set is complete and valid by using the `VALIDATE POLICYSET` command.

The `ACTIVATE POLICYSET` command fails if any of the following conditions exist:

- A copy group specifies a copy storage pool as a destination.
- A management class specifies a copy storage pool as the destination for files that were migrated by a Tivoli Storage Manager for Space Management client.
- The policy set has no default management class.
- A **TOCDESTINATION** parameter is specified, and the storage pool is either a copy pool or has a data format other than `NATIVE` or `NONBLOCK`.

The ACTIVE policy set and the last activated policy set are not necessarily identical. You can modify the original policy set that you activated without affecting the ACTIVE policy set.

If the server has data retention protection enabled, the following conditions must exist:

- All management classes in the policy set to be activated must contain an archive copy group.
- If a management class exists in the active policy set, a management class with the same name must exist in the policy set to be activated.
- If an archive copy group exists in the active policy set, the corresponding copy group in the policy set to be activated must have a `RETVER` value at least as large as the corresponding values in the active copy group.

**Attention:** Retention protection only applies to archive objects.

### 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—domain_name—policy_set_name—`►►

### Parameters

*domain\_name* **(Required)**

Specifies the policy domain for which you want to activate a policy set.

*policy\_set\_name* **(Required)**

Specifies the policy set to activate.

## ACTIVATE POLICYSET

### Example: Activate a policy set on a specific policy domain

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

```
activate policyset employee_records vacation
```

### Related commands

Table 24. Commands related to ACTIVATE POLICYSET

| Command            | Description  |
|--------------------|--|
| COPY POLICYSET     | Creates a copy of a policy set.  |
| DEFINE POLICYSET   | Defines a policy set within the specified policy domain.   |
| DELETE POLICYSET   | Deletes a policy set, including its management classes and copy groups, from a policy domain.        |
| QUERY DOMAIN       | Displays information about policy domains.   |
| QUERY POLICYSET    | Displays information about policy sets.  |
| UPDATE POLICYSET   | Changes the description of a policy set.   |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

## ASSIGN DEFMGMTCLASS (Assign a default management class)

Use this command to specify a management class as the default management class for a policy set. You must assign a default management class for a policy set before you can activate that policy set.

To ensure that clients can always back up and archive files, choose a default management class that contains both an archive copy group and a backup copy group.

The server uses the default management class to manage client files when a management class is not otherwise assigned or appropriate. For example, the server uses the default management class when a user does not specify a management class in the include-exclude list. See the *Administrator's Guide* for details.

### 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—*domain\_name*—*policy\_set\_name*—*class\_name*—►

### Parameters

#### *domain\_name* (Required)

Specifies the policy domain to which the management class belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set for which you want to assign a default management class. You cannot assign a default management class to the ACTIVE policy set.

#### *class\_name* (Required)

Specifies the management class that is to be the default management class for the policy set.

### Example: Assign a default management class

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

```
assign defmgmtclass prog1 summer default1
```

### Related commands

Table 25. Commands related to ASSIGN DEFMGMTCLASS

| Command            | Description  |
|--------------------|--|
| ACTIVATE POLICYSET | Validates and activates a policy set.  |
| DEFINE COPYGROUP   | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE MGMTCLASS   | Defines a management class.  |

## ASSIGN DEFMGMTCLASS

*Table 25. Commands related to ASSIGN DEFMGMTCLASS (continued)*

| Command            | Description  |
|--------------------|--|
| DEFINE POLICYSET   | Defines a policy set within the specified policy domain.   |
| DELETE MGMTCLASS   | Deletes a management class and its copy groups from a policy domain and policy set.                  |
| QUERY COPYGROUP    | Displays the attributes of a copy group.   |
| QUERY MGMTCLASS    | Displays information about management classes.   |
| QUERY POLICYSET    | Displays information about policy sets.  |
| UPDATE COPYGROUP   | Changes one or more attributes of a copy group.  |
| UPDATE MGMTCLASS   | Changes the attributes of a management class.  |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

---

## AUDIT commands

Use the AUDIT commands to review or examine the adequacy of the database information and the storage pool volume.

The following is a list of AUDIT commands for Tivoli Storage Manager:

- “AUDIT LIBRARY (Audit volume inventories in an automated library)” on page 34
- “AUDIT LICENSES (Audit server storage usage)” on page 36
- “AUDIT VOLUME (Verify database information for a storage pool volume)” on page 37

### AUDIT LIBRARY (Audit volume inventories in an automated library)

Use this command to audit and synchronize volume inventories in an automated library.

When the AUDIT LIBRARY command is issued on a library client, the client synchronizes its inventory with the inventory on the library manager. If the library client detects inconsistencies, it corrects them by changing the ownership of the volume on the library manager.

When the AUDIT LIBRARY command is issued on a server where the library is SCSI, 349X, or ACSLS (LIBTYPE=SCSI, LIBTYPE=349X, or LIBTYPE=ACSL), the server synchronizes its inventory with the inventory of the library device. If the server detects inconsistencies, it deletes missing volumes from its inventory.

- In SCSI libraries, the server also updates the locations of volumes in its inventory that have been moved since the last audit.
- In 349X libraries, the server also ensures that scratch volumes are in the scratch category and that private volumes are in the private category.

When the AUDIT LIBRARY command is issued on a server that is a library manager for the library (SHARED=YES), the server updates ownership of its volumes if it detects inconsistencies.

Regardless the type of server or type of library, issuing the AUDIT LIBRARY command does not automatically add new volumes to a library. To add new volumes, you must use the CHECKIN LIBVOLUME command. Note also that volumes in use by a z/OS® server using LAN-free data movement are not included in audits.

**Attention:** The following precautions apply to SCSI, 349X, and ACSLS libraries only (LIBTYPE=SCSI, LIBTYPE=349X, and LIBTYPE=ACSL):

- Running the AUDIT LIBRARY command prevents any other library activity until the audit completes. For example, Tivoli Storage Manager will not process restore or retrieve requests that involve the library when the AUDIT LIBRARY command is running.
- If other activity is occurring in the library, do not issue the AUDIT LIBRARY command. Issuing the AUDIT LIBRARY command when a library is active can produce unpredictable results (for example, a hang condition) if a process currently accessing the library attempts to acquire a new tape mount.

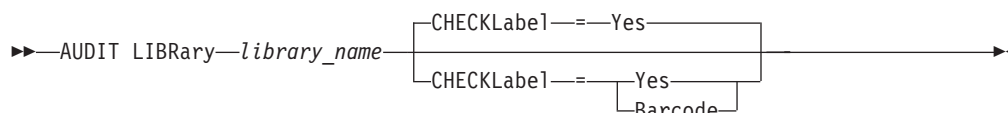
This command creates a background process that you can cancel with the CANCEL PROCESS command. To display information about background processes, use the QUERY PROCESS command.

#### Privilege class

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



## Syntax



## Parameters

### *library\_name* (Required)

Specifies the name of the library to audit.

### CHECKLabel

Specifies how the storage volume label is checked during the audit. This parameter applies to SCSI libraries only. The parameter is ignored for other library types. The default is YES. Possible values are:

#### Yes

Specifies that Tivoli Storage Manager checks each volume label to verify the identity of the volume.

#### Barcode

Specifies that Tivoli Storage Manager uses the barcode reader to read the storage label. Using the barcode decreases the audit processing time. This parameter applies only to SCSI libraries.

**Attention:** If the scanner cannot read the barcode label or the barcode label is missing, Tivoli Storage Manager loads that tape in a drive to read the label.

## Example: Audit an automated library

Audit the EZLIFE automated library.

```
audit library ezlife
```

## Related commands

Table 26. Commands related to AUDIT LIBRARY

| Command         | Description  |
|-----------------|--|
| CANCEL PROCESS  | Cancels a background server process.                         |
| DEFINE LIBRARY  | Defines an automated or manual library.                      |
| DELETE LIBRARY  | Deletes a library.   |
| DISMOUNT VOLUME | Dismounts a sequential, removable volume by the volume name. |
| QUERY LIBRARY   | Displays information about one or more libraries.            |
| QUERY LIBVOLUME | Displays information about a library volume.                 |
| QUERY PROCESS   | Displays information about background processes.             |
| UPDATE LIBRARY  | Changes the attributes of a library.                         |

## AUDIT LICENSES (Audit server storage usage)

Use this command to audit the server storage used by client nodes and to audit the server licenses. The audit determines whether the current configuration is in compliance with the license terms.

An audit creates a background process you can cancel with the CANCEL PROCESS command. If you halt and restart the server, an audit is run automatically as specified by the SET LICENSEAUDITPERIOD. To view audit results, use the QUERY LICENSE command.

**Attention:** The audit of server storage can take a lot of CPU time. You can use the AUDITSTORAGE server option to specify that storage is not to be audited.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—AUDIT LICENSES—◄◄

### Parameters

None.

### Example: Audit server licenses

Issue the AUDIT LICENSES command.

```
audit licenses
```

### Related commands

Table 27. Commands related to AUDIT LICENSES

| Command                | Description   |
|------------------------|---|
| CANCEL PROCESS         | Cancels a background server process.  |
| QUERY AUDITOCCUPANCY   | Displays the server storage utilization for a client node.                              |
| QUERY LICENSE          | Displays information about licenses and audits.   |
| QUERY PROCESS          | Displays information about background processes.  |
| QUERY STATUS           | Displays the settings of server parameters, such as those selected by the SET commands. |
| REGISTER LICENSE       | Registers a new license with the IBM Tivoli Storage Manager server.                     |
| SET LICENSEAUDITPERIOD | Specifies the number of days between automatic license audits.                          |

## AUDIT VOLUME (Verify database information for a storage pool volume)

Use this command to check for inconsistencies between database information and a storage pool volume. Processing information generated during an audit is sent to the activity log and server console.

You can only audit volumes that belong to storage pools with DATAFORMAT=NATIVE and DATAFORMAT=NONBLOCK.

You cannot audit a volume if it is being deleted from a primary or copy storage pool.

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

If the server detects a file with errors, handling of the file will depend on the type of storage pool to which the volume belongs, whether the FIX option is specified on this command, and whether the file is also stored on a volume assigned to other pools.

If Tivoli Storage Manager does not detect errors for a file that was marked as damaged, the state of the file is reset so that it can be used.

The Tivoli Storage Manager server will not delete archive files that are on deletion hold. If archive retention protection is enabled, the Tivoli Storage Manager server will not delete archive files whose retention period has not expired.

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

To audit multiple volumes, you can use the FROMDATE and TODATE parameters. Use the STGPOOL parameter to audit all volumes in a storage pool. When you use the parameters FROMDATE, TODATE, or both, the server limits the audit to only the sequential media volumes that meet the date criteria, and automatically includes all online disk volumes in storage. To limit the number of volumes that may include disk volumes, use the FROMDATE, TODATE, and STGPOOL parameters.

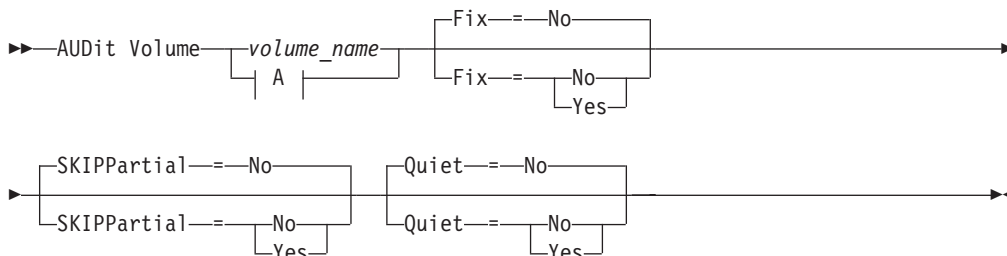
If you are running a server with archive retention protection enabled, and you have data stored in storage pools which are defined with the parameter RECLAMATIONTYPE=SNAPLOCK, the Last Access Date on the NetApp SnapLock Filer for a volume should be equal to the End Reclaim Period date that you see when you issue a QUERY VOLUME F=D command on that volume. During AUDIT VOLUME processing, these dates are compared. If they do not match and the AUDIT VOLUME command is being run with the FIX=NO parameter, a message will be issued to you indicating that the command should be run with the FIX=YES parameter to resolve the inconsistency. If they do not match and the AUDIT VOLUME command is being run with the FIX=YES parameter, the inconsistencies will be resolved.

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.

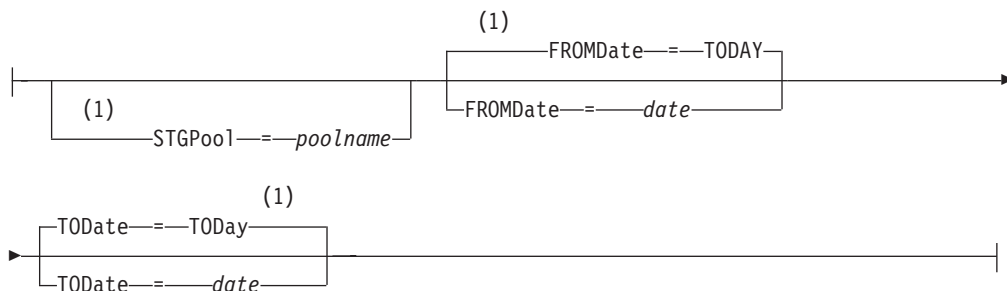
## Privilege class

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.

## Syntax



**A (at least one of these parameters must be specified):**



### Notes:

- 1 You cannot specify a volume name if you specify a storage pool name, FROMDATE, or TODATE.

## Parameters

### *volume\_name*

Specifies the name of the storage pool volume you want to audit. This parameter is required if you do not specify a storage pool. You cannot specify a volume name together with the FROMDATE and TODATE parameters.

### Fix

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

The actions the server performs depend on whether the volume is assigned to a primary or a copy storage pool.

### Primary Storage Pool:

**Note:** If the AUDIT VOLUME command does not detect an error in a file that was previously marked as damaged, Tivoli Storage Manager 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 a correctable hardware problem such as a dirty tape head.

**Fix=No**

Tivoli Storage Manager reports, but does not delete, database records that refer to files with inconsistencies:

- Tivoli Storage Manager marks the file as damaged in the database. If a backup copy is stored in a copy storage pool, you can restore the file using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the file is a cached copy, you must delete references to the file on this volume by issuing the AUDIT VOLUME command and specifying FIX=YES. If the physical file is not a cached copy, and a duplicate is stored in a copy storage pool, it can be restored by using the RESTORE VOLUME or RESTORE STGPOOL command.

**Fix=Yes**

The server fixes any inconsistencies as they are detected:

- If the physical file is a cached copy, the server deletes the database records that refer to the cached file. The primary file is stored on another volume.
- If the physical file is not a cached copy, and the file is also stored in one or more copy storage pools, the error will be reported and the physical file marked as damaged in the database. You can restore the physical file by using the RESTORE VOLUME or RESTORE STGPOOL command.
- If the physical file is not a cached copy, and the physical file is not stored in a copy storage pool, each logical file for which inconsistencies are detected are deleted from the database.
- If archive retention protection is enabled by using the SET ARCHIVERETENTIONPROTECTION command, a cached copy of data can be deleted if needed. Data in primary and copy storage pools can only be marked damaged and never deleted.

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.

**Copy Storage Pool:**
**Fix=No**

The server reports the error and marks the physical file copy as damaged in the database.

**Fix=Yes**

The server deletes any references to the physical file and any database records that point to a physical file that does not exist.

**SKIPPARTIAL**

Specifies whether Tivoli Storage Manager ignores skipped files, which are files that span multiple storage pool volumes. 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**

Tivoli Storage Manager audits files that span multiple volumes.

Unless you specify SKIPPARTIAL=YES, Tivoli Storage Manager attempts to process each file stored on the volume, including files that span into and out of other volumes. 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.

### Yes

Tivoli Storage Manager audits only files that are stored on the volume to be audited. The status of any skipped files is unknown.

### Quiet

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

### No

Specifies that Tivoli Storage Manager sends detailed informational messages and a summary. Each message contains the node, file space, and client name for the file.

### Yes

Specifies that Tivoli Storage Manager sends only a summary report.

### FROMDate

Specifies the beginning date of the range to audit volumes. The default is the current date. All sequential media volumes meeting the time range criteria that were written to after this date are audited. The server includes all online disk volumes in storage. The server starts one audit process for each volume and runs the process serially. You cannot use this parameter if you have specified a volume. This parameter is optional. To limit the number of volumes that may include disk volumes, use the FROMDATE, TODATE, and STGPOOL parameters.

You can specify the date by using one of the following values:

| Value               | Description  | Example  |
|---------------------|--|--|
| MM/DD/YYYY          | A specific date  | 10/15/2001<br><br>If a date is entered, all candidate volumes written on that day (starting at 12:00:01 am) will be evaluated.             |
| TODAY               | The current date   | TODAY  |
| TODAY-days or -days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -7 or -7.<br><br>To display information beginning with volumes written a week ago, you can specify FROMDATE=TODAY-7 or FROMDATE= -7. |

### TODate

Specifies the ending date of the range for volumes to audit. All sequential media volumes meeting the time range criteria that were written to before this date are audited. The server includes all online disk volumes in storage. If you do not specify a value, the server defaults to the current date. You cannot use this parameter if you have specified a volume. This parameter is optional. To limit the number of volumes that may include disk volumes, use the FROMDATE, TODATE, and STGPOOL parameters.

You can specify the date by using one of the following values:

| Value                  | Description  | Example  |
|------------------------|--|--|
| <i>MM/DD/YYYY</i>      | A specific date  | 10/15/2001<br><br>If a date is entered, all candidate volumes written on that day (ending at 11:59:59 pm) will be evaluated. |
| TODAY                  | The current date   | TODAY  |
| TODAY-days or<br>-days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY-1 or -1.<br><br>To display information created up to yesterday, you can specify TODATE=TODAY-1 or simply TODATE= -1.   |

### STGPool

This parameter specifies that the server only audits the volumes from the specified storage pool. This parameter is optional. You cannot use this parameter if you have specified a volume.

### Example: Verify database information for a specific storage pool volume

Verify that the database information for storage pool volume PROG2 is consistent with the data stored on the volume. Tivoli Storage Manager fixes any inconsistencies.

```
audit volume prog2 fix=yes
```

### Example: Verify database information for all volumes written to during a specific date range

Verify that the database information for all eligible volumes written to from 3/20/2002 to 3/22/2002 is consistent with data stored on the volume.

```
audit volume fromdate=03/20/2002 todate=03/22/2002
```

### Example: Verify database information for all volumes in a specific storage pool

Verify that the database information for all volumes in storage pool STPOOL3 is consistent with data stored on the volume for today.

```
audit volume stgpool=STPOOL3
```

### Example: Verify database information for all volumes in a specific storage pool written to in the last two days

Verify that the database information for all volumes in storage pool STPOOL3 is consistent with data stored on the volume for the last two days.

```
audit volume stgpool=STPOOL3 fromdate=-1
```

## Related commands

Table 28. Commands related to AUDIT VOLUME

| Command        | Description                          |
|----------------|--------------------------------------|
| CANCEL PROCESS | Cancels a background server process. |

## AUDIT VOLUME

*Table 28. Commands related to AUDIT VOLUME (continued)*

| Command                        | Description  |
|--------------------------------|--|
| QUERY CONTENT                  | Displays information about files in a storage pool volume. |
| QUERY PROCESS                  | Displays information about background processes.           |
| QUERY VOLUME                   | Displays information about storage pool volumes.           |
| SET ARCHIVERETENTIONPROTECTION | Specifies whether data retention protection is activated.  |



---

## BACKUP commands

Use the BACKUP command to create backup copies of Tivoli Storage Manager information or objects.

The following are the BACKUP commands for Tivoli Storage Manager:

- “BACKUP DB (Back up the database)” on page 44
- “BACKUP DEVCONFIG (Create backup copies of device configuration information)” on page 47
- “BACKUP NODE (Back up a NAS node)” on page 49
- “BACKUP STGPOOL (Back up primary storage pool to copy storage pool)” on page 54
- “BACKUP VOLHISTORY (Save sequential volume history information)” on page 58

### BACKUP DB (Back up the database)

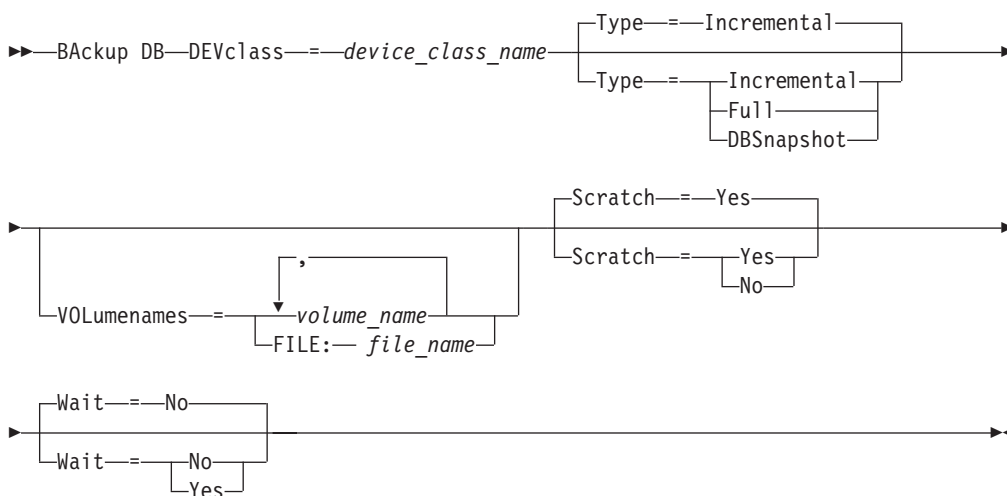
Use this command to back up a Tivoli Storage Manager database to sequential access volumes.

To determine how much additional storage space a backup requires, use the QUERY DB command.

#### Privilege class

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

#### Syntax



#### Parameters

##### DEVclass (Required)

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

##### Restriction:

- You cannot use a device class with a device type of NAS or CENTERA.
- A restore database operation fails if the source for the restore is a FILE library. A FILE library is created if the FILE device class specifies `SHARED=YES`.

If all drives for this device class are busy when the backup runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available for the backup.

##### Type

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

##### Incremental

Specifies that you want to run an incremental backup of the Tivoli Storage Manager database. An incremental (or cumulative) backup image contains a copy of all database data that has changed since the last successful full backup operation was performed.

**Full**

Specifies that you want to run a full backup of the Tivoli Storage Manager database.

**DBSnapshot**

Specifies that you want to run a full snapshot database backup. The entire contents of a database are copied and a new snapshot database backup is created without interrupting the existing full and incremental backup series for the database.

**VOLumenames**

Specifies the volumes used to back up the database. This parameter is optional. However, if you specify `SCRATCH=NO`, you must specify a list of volumes.

*volume\_name*

Specifies the volumes used to back up the database. Specify multiple volumes by separating the names with commas and no intervening spaces.

**FILE:filename**

Specifies the name of a file that contains a list of volumes used to back up the database. Each volume name must be on a separate line. Blank lines and comment lines, which begin with an asterisk, are ignored.

For example, to use volumes DB0001, DB0002, and DB0003, create a file that contains these lines:

```
DB0001
DB0002
DB0003
```

Name the file appropriately. For example:

```
TAPEVOL
```

You can then specify the volumes for the command as follows:

```
VOLUMENAMES=FILE:TAPEVOL
```

**Scratch**

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

**Yes**

Specifies that scratch volumes can be used.

If you specify `SCRATCH=YES` and the `VOLUMENAMES` parameter, Tivoli Storage Manager only uses scratch volumes if space is unavailable 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 use the default.

**No**

Specifies that scratch volumes cannot be used.

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

**Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. The default is NO. Possible values are:

### No

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use 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.

### Yes

Specifies that the server processes this command in the foreground. Wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

## Example: Run an incremental backup using a scratch volume

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

```
backup db devclass=file type=incremental
```

## Related commands

Table 29. Commands related to BACKUP DB

| Command                   | Description   |
|---------------------------|---|
| BACKUP DEVCONFIG          | Backs up IBM Tivoli Storage Manager device information to a file.                     |
| BACKUP VOLHISTORY         | Records volume history information in external files.                                 |
| CANCEL PROCESS            | Cancels a background server process.  |
| DELETE VOLHISTORY         | Removes sequential volume history information from the volume history file.           |
| EXPIRE INVENTORY          | Manually starts inventory expiration processing.                                      |
| MOVE DRMEDIA              | Moves DRM media on-site and off-site.   |
| PREPARE                   | Creates a recovery plan file.   |
| QUERY DB                  | Displays allocation information about the database.                                   |
| QUERY PROCESS             | Displays information about background processes.                                      |
| QUERY VOLHISTORY          | Displays sequential volume history information that has been collected by the server. |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.                             |

## BACKUP DEVCONFIG (Create backup copies of device configuration information)

Use this command to back up information about device configuration for the server. To restore the Tivoli Storage Manager database, device configuration information must be available.

This command backs up the following information in one or more files:

- Device class definitions
- Library definitions
- Drive definitions
- Path definitions when SRCTYPE=SERVER
- Server definitions
- Server name
- Server password
- Volume location information for LIBTYPE=SCSI libraries

You can use the DEVCONFIG server option to specify one or more files in which to store device configuration information. Tivoli Storage Manager updates the files whenever a device class, library, or drive is defined, updated, or deleted.

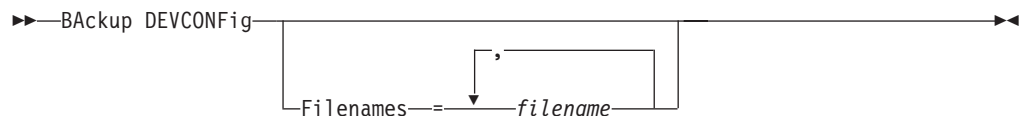
To ensure updates are complete before the server is halted:

- Do not halt the server for a few minutes after issuing the BACKUP DEVCONFIG command.
- Specify multiple DEVCONFIG options in the server options file.
- Examine the device configuration file to see if the file has been updated.

### Privilege class

Any administrator can issue this command unless it includes the FILENAMES parameter. If the FILENAMES parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES, the administrator must have system privilege. If the FILENAMES parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO, the administrator must have operator, policy, storage or system privilege.

### Syntax



### Parameters

#### FileNames

Specifies the files in which to store device configuration information. You can specify multiple files by separating the names with commas and no intervening spaces. This parameter is optional.

If you do not specify a file name, Tivoli Storage Manager stores the information in all files specified with the DEVCONFIG option in the server options file.

## Example: Backup device configuration information to a file

Back up device configuration information to a file named DEVICE.

```
backup devconfig filenames=device
```

## Related commands

Table 30. Commands related to BACKUP DEVCONFIG

| Command            | Description   |
|--------------------|---|
| CHECKIN LIBVOLUME  | Checks a storage volume into an automated library.    |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library.  |
| DEFINE DEVCLASS    | Defines a device class.                               |
| DEFINE DRIVE       | Assigns a drive to a library.                         |
| DEFINE LIBRARY     | Defines an automated or manual library.               |
| DEFINE PATH        | Defines a path from a source to a destination.        |
| DEFINE SERVER      | Defines a server for server-to-server communications. |
| LABEL LIBVOLUME    | Labels volumes in manual or automated libraries.      |
| QUERY LIBVOLUME    | Displays information about a library volume.          |
| SET SERVERNAME     | Specifies the name by which the server is identified. |
| SET SERVERPASSWORD | Specifies the server password.                        |
| UPDATE DEVCLASS    | Changes the attributes of a device class.             |
| UPDATE DRIVE       | Changes the attributes of a drive.                    |
| UPDATE LIBRARY     | Changes the attributes of a library.                  |
| UPDATE LIBVOLUME   | Changes the status of a storage volume.               |
| UPDATE PATH        | Changes the attributes associated with a path.        |
| UPDATE SERVER      | Updates information about a server.                   |

## BACKUP NODE (Back up a NAS node)

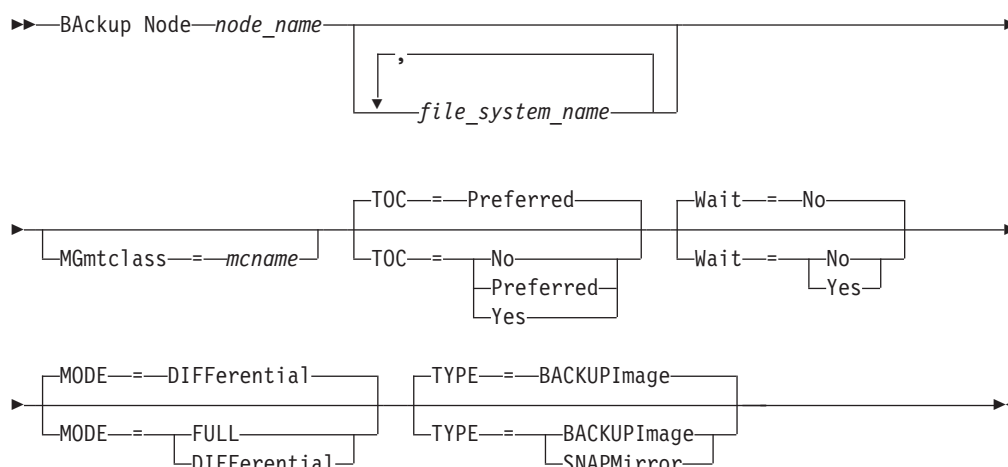
Use this command to start a backup operation for a network-attached storage (NAS) node.

Backups that are created for NAS nodes with this BACKUP NODE command are functionally equivalent to backups that are created by using the BACKUP NAS command on a Tivoli Storage Manager client. You can restore these backups with either the server's RESTORE NODE command or the client's RESTORE NAS command.

### Privilege class

To issue this command, you must have system privilege, policy privilege for the domain to which the node is assigned, or client owner authority over the node.

### Syntax



### Parameters

#### *node\_name* (Required)

Specifies the node for which the backup will be performed. You cannot use wildcard characters or specify a list of names.

#### *file\_system\_name*

Specifies the name of one or more file systems to back up. You can also specify names of virtual file spaces that have been defined for the NAS node. The file system name that you specify cannot contain wildcard characters. You can specify more than one file system by separating the names with commas and no intervening spaces.

If you do not specify a file system, all file systems will be backed up. Any virtual file spaces defined for the NAS node are backed up as part of the file system image, not separately.

If a file system exists on the NAS device with the same name as the virtual file space specified, Tivoli Storage Manager automatically renames the existing virtual file space in the server database, and backs up the NAS file system which matches the name specified. If the virtual file space has backup data, the file space definition associated with the virtual file space will also be renamed.

**Tip:** See the virtual file space name parameter in the DEFINE VIRTUALFSMAPPING command for more naming considerations.

In determining the file systems to process, the server will not use any DOMAIN.NAS, INCLUDE.FS.NAS, or EXCLUDE.FS.NAS statements in any client option file or client option set. If you back up multiple file systems, the backup of each file system is a separate server process.

### **MGmtclass**

Specifies the name of the management class to which this backup data is bound. If you do not specify a management class, the backup data is bound to the default management class of the policy domain to which the node is assigned. In determining the management class, the server will *not* use any INCLUDE.FS.NAS statements in any client option file or client option set. The destination management class may refer to a Tivoli Storage Manager native pool, in which case Network Data Management Protocol (NDMP) data is sent into the Tivoli Storage Manager native hierarchy. After this occurs, the data stays in the Tivoli Storage Manager hierarchy. Data flowing to Tivoli Storage Manager native pools goes over the LAN and data flowing to NAS pools can be directly attached or over a SAN.

### **TOC**

Specifies whether a table of contents (TOC) is saved for each file system backup. Consider the following in determining whether you want to save a table of contents.

- If a table of contents is saved, you will be able to use the QUERY TOC command to determine the contents of a file system backup in conjunction with the RESTORE NODE command to restore individual files or directory trees. You will also be able to use the Tivoli Storage Manager Web backup-archive client to examine the entire file system tree and choose files and directories to restore. Creation of a table of contents requires that you define the TOCDESTINATION attribute in the backup copy group for the management class to which this backup image is bound. Note that a table of contents creation requires additional processing, network resources, storage pool space, and possibly a mount point during the backup operation.
- A table of contents for a NAS file system cannot have a directory path greater than 1024 characters.
- If a table of contents is not saved for a file system backup, you will still be able to restore individual files or directory trees using the RESTORE NODE command, provided that you know the fully qualified name of each file or directory to be restored and the image in which that object was backed up. This parameter is optional. The default value is Preferred. Possible values are:

#### **No**

Specifies that table of contents information is not saved for file system backups.

#### **Preferred**

Specifies that table of contents information should be saved for file system backups. However, a backup does not fail just because an error occurs during creation of the table of contents. This is the default value.

#### **Yes**

Specifies that table of contents information must be saved for each file system backup. A backup fails if an error occurs during creation of the table of contents.



**Attention:** If **MODE=DIFFERENTIAL** is specified and a table of contents is requested (**TOC=PREFERRED** or **TOC=YES**), but the last full image does not have a table of contents, a full backup will be performed and a table of contents will be created for that full backup.

## Wait

Specifies whether to wait for the server to complete processing this command in the foreground. The default is **NO**. Possible values are:

### No

Specifies that the server processes this command in the background. Use the **QUERY PROCESS** command to monitor the background processing of this command.

### Yes

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes. If you are backing up multiple file systems, all backup processes must complete before the command is complete.

**Attention:** You cannot specify **WAIT=YES** from the server console.

## MODE

Specifies whether the file system backups are full or differential. The default is **DIFFERENTIAL**.

### FULL

Specifies to back up the entire file system.

### DIFFerential

Specifies that only the files that have changed since the most recent full backup should be backed up. If you choose a differential backup, and a full backup is not found, a full backup is performed. You cannot specify **TYPE=SNAPMIRROR** when the **MODE** parameter is set to **DIFFERENTIAL**.

## TYPE

Specifies the backup method used to perform the NDMP backup operation. The default value for this parameter is **BACKUPIIMAGE** and it should be used to perform a standard NDMP base or differential backup. Other image types represent backup methods that might be specific to a particular file server. Possible values are:

### BACKUPIImage

Specifies that the file system should be backed up using an NDMP dump operation. This is the default method for performing an NDMP backup. The **BACKUPIIMAGE** type operation supports full and differential backups, file-level restore processing and directory-level backup.

### SNAPMirror

Specifies that the file system should be copied to a Tivoli Storage Manager storage pool using the NetApp SnapMirror to Tape function. SnapMirror images are block level full backup images of a file system. Typically, a SnapMirror backup takes significantly less time to perform than a traditional NDMP full file system backup. However there are limitations and restrictions on how SnapMirror images can be used. The SnapMirror to Tape function is intended to be used as a disaster-recovery option for copying very large NetApp file systems to secondary storage.

## BACKUP NODE

For most NetApp file systems, use the standard NDMP full or differential backup method. See the *Administrator's Guide* for limitations on using SnapMirror images as a backup method.

Refer to the documentation that came with your NetApp file server for more information. When setting the TYPE parameter to SNAPMirror, note the following restrictions:

1. You cannot specify TOC=YES or TOC=PREFERRED.
2. The file\_system\_name cannot be a virtual filesystem name.
3. The snapshot which is created automatically by the file server during the SnapMirror copy operation will be deleted at end of the operation.
4. This parameter is valid for NetApp and IBM N-Series file servers only.

### Example: Perform a full backup

Perform a full backup on the /vol/vol10 file system of NAS node NAS1.

```
backup node nas1 /vol/vol10 mode=full
```

### Example: Perform a backup on a directory and create a table of contents

Back up the directory /vol/vol2/mikes on the node NAS1 and create a table of contents for the image. For the following two examples, assume Table 31 contains the virtual file space definitions exist on the server for the node NAS1.

```
backup node nas1 /mikesdir
```

Table 31. Virtual file space definitions

| Virtual file space name | File system | Path           |
|-------------------------|-------------|----------------|
| /mikesdir               | /vol/vol2   | /mikes         |
| /DataDirVol2            | /vol/vol2   | /project1/data |
| /TestDirVol1            | /vol/vol1   | /project1/test |

### Example: Perform a backup on two directories

Back up the directories /vol/vol2/project1/data and /vol/vol1/project1/test of the node NAS1. Refer to Table 31 for the the virtual file space definitions that exist on the server for the node NAS1.

```
backup node nas1 /DataDirVol2,/testdirvol1 mode=full toc=yes
```

### Related commands

Table 32. Commands related to BACKUP NODE

| Command  | Description  |
|--|--|
| BACKUP NAS (Tivoli Storage Manager client command) | Creates a backup of NAS node data.   |
| DEFINE COPYGROUP                                   | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE VIRTUALFSMAPPING                            | Define a virtual file space mapping.   |
| QUERY NASBACKUP                                    | Displays information about NAS backup images.  |

*Table 32. Commands related to BACKUP NODE (continued)*

| <b>Command</b>                                      | <b>Description</b>   |
|---|--|
| QUERY TOC   | Displays details about the table of contents for a specified backup image. |
| QUERY COPYGROUP                                     | Displays the attributes of a copy group.                                   |
| RESTORE NAS (Tivoli Storage Manager client command) | Restores a backup of NAS node data.  |
| RESTORE NODE  | Restores a network-attached storage (NAS) node.                            |
| UPDATE COPYGROUP                                    | Changes one or more attributes of a copy group.                            |

## BACKUP STGPOOL (Back up primary storage pool to copy storage pool)

Use this command to back up primary storage pool files to a copy storage pool.

In addition to backing up primary storage pools having NATIVE or NONBLOCK data formats, this command lets you back up primary storage pools that have NDMP data formats (NETAPPDUMP, CELERRADUMP, or NDMPDUMP). The copy storage pool to which data is to be backed up must have the same data format as the primary storage pool. Tivoli Storage Manager supports backend data movement for NDMP images. For details, see the *Administrator's Guide*.

You cannot back up data from or to storage pools defined with a CENTERA device class.

If a file already exists in the copy storage pool, the file is not backed up unless the copy of the file in the copy storage pool is marked as damaged. However, a new copy is not created if the file in the primary storage pool is also marked as damaged. In a random-access storage pool, neither cached copies of migrated files nor damaged primary files are backed up.

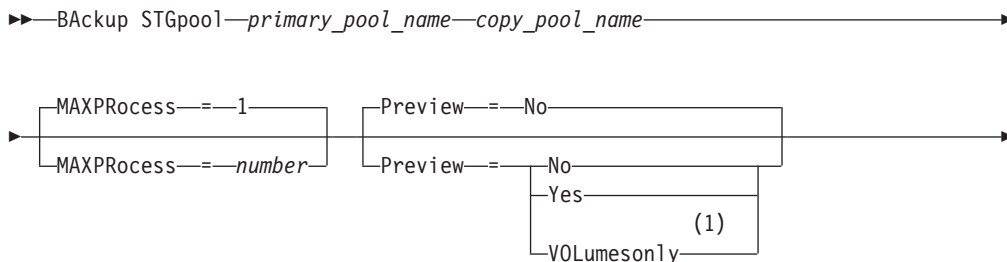
If migration for a storage pool starts during a storage pool backup, some files may be migrated before they are backed up. You may want to back up storage pools that are higher in the migration hierarchy before backing up storage pools that are lower. For example, when performing a storage pool backup to copy the contents of a storage pool offsite, if the process is not done according to the existing storage pool hierarchy, some files may not be copied to the copy storage pool. This could become critical for disaster recovery purposes. When performing a storage pool backup on multiple storage pools, the primary storage pool should be completed before starting the backup process on the next storage pool.

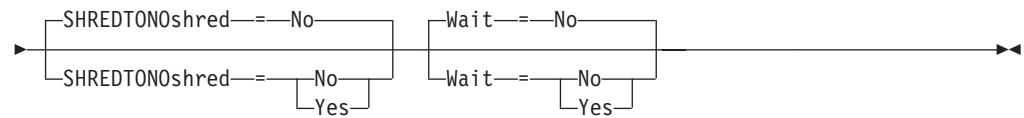
**Remember:** Issuing this command for a primary storage pool that is set up for deduplication removes duplicate data, if the copy storage pool is also set up for deduplication.

## 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





## Notes:

- 1 Valid only for storage pools associated with a sequential-access device class.

## Parameters

### *primary\_pool* (Required)

Specifies the primary storage pool.

### *copy\_pool* (Required)

Specifies the copy storage pool.

### MAXProcess

Specifies the maximum number of parallel processes to use for backing up files. This parameter is optional. Enter a value from 1 to 999. The default is 1.

Using multiple, parallel processes may improve throughput for the backup. The expectation is that the time needed to perform the storage pool backup will be decreased by using multiple processes. However, when multiple processes are running, in some cases one or more of the processes needs to wait to use a volume that is already in use by a different backup process.

When determining this value, consider the number of logical and physical drives that can be dedicated to this operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the backup.

Each process needs a mount point for copy storage pool volumes, and, if the device type is not FILE, each process also needs a drive. If you are backing up a sequential storage pool, each process needs an additional mount point for primary storage pool volumes and, if the device type is not FILE, an additional drive. For example, suppose you specify a maximum of 3 processes to back up a primary sequential storage pool to a copy storage pool of the same device class. Each process requires 2 mount points and 2 drives. To run all 3 processes, the device class must have a mount limit of at least 6, and at least 6 mount points and 6 drives must be available.

To preview a backup, only one process is used and no mount points or drives are needed.

### Preview

Specifies if you want to preview but not perform the backup. The preview displays the number of files and bytes to be backed up and a list of the primary storage pool volumes that you must mount. This parameter is optional. The default is NO. Possible values are:

#### No

Specifies that the backup is done.

#### Yes

Specifies that you want to preview the backup but not do the backup.

### VOLumesonly

Specifies that you want to preview the backup only as a list of the volumes

that must be mounted. This choice requires the least processing time. The VOLUMESONLY option is valid only for storage pools associated with a sequential-access device class.

### SHREDTONOshred

Specifies whether data will be backed up to a copy storage pool from a primary storage pool that enforces shredding. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that the server will not allow data to be backed up to a copy storage pool from a primary storage pool that enforces shredding. If the primary storage pool enforces shredding, the operation will fail.

#### Yes

Specifies that the server does allow data to be backed up to a copy storage pool from a primary storage pool that enforces shredding. The data in the copy storage pool will not be shredded when it is deleted.

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is NO. Possible values are:

#### No

Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed. Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files may have already been backed up prior to the cancellation.

#### Yes

Specifies that the server performs this operation in the foreground. You must wait for the operation to complete before continuing with other tasks. The server displays the output messages to the administrative client when the operation completes.

**Note:** You cannot specify WAIT=YES from the server console.

## Example: Backup the primary storage pool

Back up the primary storage pool named PRIMARY\_POOL to the copy storage pool named COPYSTG.

```
backup stgpool primary_pool copystg
```

## Related commands

Table 33. Commands related to BACKUP STGPOOL

| Command        | Description   |
|----------------|---|
| CANCEL PROCESS | Cancels a background server process.                  |
| MOVE DRMEDIA   | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA  | Displays information about disaster recovery volumes. |

*Table 33. Commands related to BACKUP STGPOOL (continued)*

| Command           | Description   |
|-------------------|---|
| QUERY PROCESS     | Displays information about background processes.  |
| QUERY SHREDSTATUS | Displays information about data waiting to be shredded.                                       |
| QUERY STGPOOL     | Displays information about storage pools.   |
| RESTORE STGPOOL   | Restores files to a primary storage pool from copy storage pools.                             |
| RESTORE VOLUME    | Restores files stored on specified volumes in a primary storage pool from copy storage pools. |
| SHRED DATA        | Manually starts the process of shredding deleted data.  |

## BACKUP VOLHISTORY (Save sequential volume history information)

Use this command to back up sequential volume history information to one or more files.

**Note:** You must use volume history information when you reload the database and audit affected storage pool volumes. If you cannot start the server, you can use the volume history file to query the database about these volumes.

The volume history includes information about the following types of volumes:

- Archive log volumes
- Database backup volumes
- Export volumes
- Backup set volumes
- Database snapshot volumes
- Database recovery plan file volumes
- Recovery plan file volumes
- Recovery plan file snapshot volumes
- The following sequential access storage pool volumes:
  - Volumes added to storage pools
  - Volumes reused through reclamation or MOVE DATA operations
  - Volumes removed by using the DELETE VOLUME command or during reclamation of scratch volumes

You must use the `VOLUMEHISTORY` server option to specify one or more volume history files. Tivoli Storage Manager updates volume history files whenever server sequential volume history information is changed.

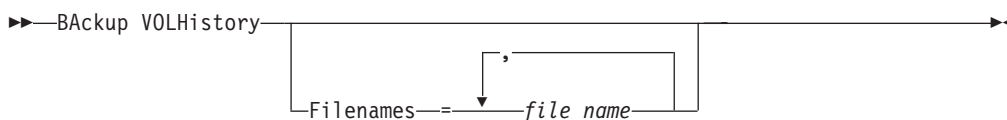
In order to ensure updates are complete before the server is halted, we recommend you:

- Not halt the server for a few minutes after issuing the `BACKUP VOLHISTORY` command.
- Specify multiple `VOLUMEHISTORY` options in the server options file.
- Examine the volume history file to see if the file has been updated.

## Privilege class

Any administrator can issue this command unless it includes the FILENAMES parameter. If the FILENAMES parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES, the administrator must have system privilege. If the FILENAMES parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO, the administrator must have operator, policy, storage or system privilege.

## Syntax





## Parameters

### Filenames

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

If you do not specify a file name, Tivoli Storage Manager stores the information in all files specified with the VOLUMEHISTORY option in the server options file.

### Example: Back up the volume history information to a file

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

```
backup volhistory filenames=volhist
```

### Related commands

Table 34. Commands related to BACKUP VOLHISTORY

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

### BEGIN EVENTLOGGING (Begin logging events)

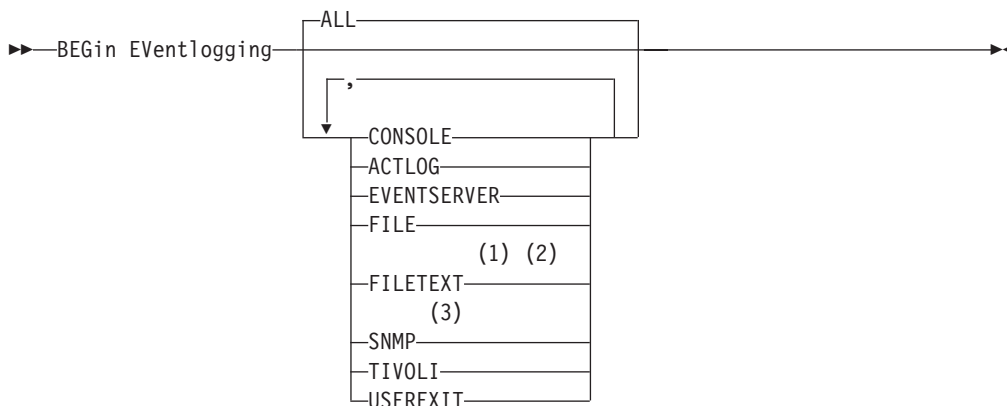
Use this command to begin logging events to one or more receivers. A receiver for which event logging has begun is an *active receiver*.

When the server is started, event logging automatically begins for the console and activity log and for any receivers that are started automatically based on entries in the server options file. You can use this command to begin logging events to receivers for which event logging is *not* automatically started at server startup. You can also use this command after you have disabled event logging to one or more receivers.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax



#### Notes:

- 1 This parameter is only available for z/OS operating system.
- 2 This parameter is only available for Windows operating system.
- 3 This parameter is only available for AIX, HP-UX, Linux, Solaris, Windows operating systems.

#### Parameters

Specify one or more receivers. You can specify multiple receivers by separating them with commas and no intervening spaces. If you specify ALL, logging begins for all receivers that are configured. The default is ALL.

##### ALL

Specifies all receivers that are configured for event logging.

##### CONSOLE

Specifies the server console as a receiver.

##### ACTLOG

Specifies the Tivoli Storage Manager activity log as a receiver.

##### EVENTSERVER

Specifies the event server as a receiver.

## FILE

Specifies a user file as a receiver. Each logged event is a record in the file and a person cannot read each logged event easily.

## FILETEXT

Specifies a user file as a receiver. Each logged event is a fixed-size, readable line.

## SNMP

Specifies the simple network management protocol (SNMP) as a receiver.

## TIVOLI

Specifies the Tivoli Management Environment® (TME®) as a receiver.

## USEREXIT

Specifies a user-written routine to which Tivoli Storage Manager writes information as a receiver.

## Example: Begin logging events

Begin logging events to the Tivoli Storage Manager activity log.

```
begin eventlogging actlog
```

## Related commands

Table 35. Commands related to BEGIN EVENTLOGGING

| Command          | Description   |
|------------------|---|
| DISABLE EVENTS   | Disables specific events for receivers.   |
| ENABLE EVENTS    | Enables specific events for receivers.  |
| END EVENTLOGGING | Ends event logging to a specified receiver.   |
| QUERY ENABLED    | Displays enabled or disabled events for a specific receiver.                            |
| QUERY EVENTRULES | Displays information about rules for server and client events.                          |
| QUERY STATUS     | Displays the settings of server parameters, such as those selected by the SET commands. |

---

## CANCEL commands

Use the CANCEL commands to end a task before it is completed.

The following is a list of CANCEL commands for Tivoli Storage Manager:

- “CANCEL EXPIRATION (Cancel an expiration process)” on page 63
- “CANCEL EXPORT (Delete a suspended export operation)” on page 64
- “CANCEL PROCESS (Cancel an administrative process)” on page 65
- “CANCEL REQUEST (Cancel one or more mount requests)” on page 67
- “CANCEL RESTORE (Cancel a restartable restore session)” on page 68
- “CANCEL SESSION (Cancel one or more client sessions)” on page 69

## CANCEL EXPIRATION (Cancel an expiration process)

Use this command to cancel a process that is running as a result of an inventory expiration operation.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—cancel expiration—◀◀

### Example: Cancel an inventory expiration process

Cancel the process that was generated by an inventory expiration operation.  
cancel expiration

### Related commands

Table 36. Command related to CANCEL EXPIRATION

| Command          | Description                                      |
|------------------|--|
| QUERY PROCESS    | Displays information about background processes. |
| EXPIRE INVENTORY | Manually starts inventory expiration processing. |

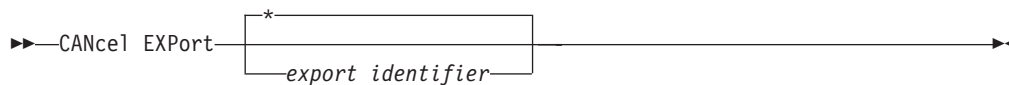
## CANCEL EXPORT (Delete a suspended export operation)

Use this command to delete a suspended server-to-server export operation. After issuing the CANCEL EXPORT command, you cannot restart the export operation. Issue the CANCEL PROCESS command to delete a currently running export operation.

### Privilege class

You must have system privilege to issue this command.

### Syntax



### Parameters

#### *export\_identifier*

The unique identifier of the suspended export operation that you wish to delete. You can also enter wildcard characters for the identifier. Issue the QUERY EXPORT command to list the currently suspended export operations.

### Example: Delete a specific suspended export operation

Cancel the suspended server-to-server export operation EXPORTALLACCTNODES.

```
cancel export exportallacctnodes
```

### Example: Delete all suspended server-to-server export operations

Cancel all suspended server-to-server export processes.

```
cancel export *
```

### Related commands

Table 37. Commands related to CANCEL EXPORT

| Command        | Description   |
|----------------|---|
| CANCEL PROCESS | Cancels a background server process.                                    |
| EXPORT NODE    | Copies client node information to external media.                       |
| EXPORT SERVER  | Copies all or part of the server to external media.                     |
| QUERY EXPORT   | Displays the export operations that are currently running or suspended. |
| RESTART EXPORT | Restarts a suspended export operation.                                  |
| SUSPEND EXPORT | Suspends a running export operation.                                    |

## CANCEL PROCESS (Cancel an administrative process)

Use this command to cancel a background process started by an administrative command or by a process, such as storage pool migration.

The following commands generate background processes:

- AUDIT LIBRARY
- AUDIT LICENSES
- AUDIT VOLUME
- BACKUP DB
- BACKUP NODE
- BACKUP STGPOOL
- CHECKIN LIBVOLUME
- CHECKOUT LIBVOLUME
- DELETE FILESPACE
- DELETE VOLUME
- EXPIRE INVENTORY
- EXPORT ADMIN
- EXPORT NODE
- EXPORT POLICY
- EXPORT SERVER
- GENERATE BACKUPSET
- IMPORT ADMIN
- IMPORT NODE
- IMPORT POLICY
- IMPORT SERVER
- MIGRATE STGPOOL
- MOVE DATA
- MOVE DRMEDIA
- MOVE MEDIA
- PREPARE
- RECLAIM STGPOOL
- RESTORE NODE
- RESTORE STGPOOL
- RESTORE VOLUME
- VARY

The following internal server operations generate background processes:

- Inventory expiration
- Migration
- Reclamation

To cancel a process, you must have the process number, which you can obtain by issuing the QUERY PROCESS command.

Some processes, such as reclamation, will generate mount requests in order to complete processing. If a process has a pending mount request, the process may not respond to a CANCEL PROCESS command until the mount request has been answered or cancelled by using either the REPLY or CANCEL REQUEST command, or by timing out.

Issue the QUERY REQUEST command to list open requests, or query the activity log to determine if a given process has a pending mount request. A mount request indicates that a volume is needed for the current process, but the volume is not available in the library. It may not be available if the administrator has issued the MOVE MEDIA or CHECKOUT LIBVOLUME command, or manually removed the volume from the library.

After you issue a CANCEL PROCESS command for an export operation, the process cannot be restarted. To stop a server-to-server export operation but allow it to be restarted at a later time, issue the SUSPEND EXPORT command.

### Privilege class

To issue this command, you must have system privilege.

## CANCEL PROCESS

### Syntax

►►—CANCEL PROCESS—*process\_number*—◄◄

### Parameters

#### *process\_number* (Required)

Specifies the number of the background process you want to cancel.

### Example: Cancel a background process using its process number

Cancel background process number 3.

```
cancel process 3
```

### Related commands

*Table 38. Commands related to CANCEL PROCESS*

| Command        | Description   |
|----------------|---|
| CANCEL EXPORT  | Deletes a suspended export operation                                    |
| CANCEL REQUEST | Cancels pending volume mount requests.                                  |
| QUERY EXPORT   | Displays the export operations that are currently running or suspended. |
| QUERY PROCESS  | Displays information about background processes.                        |
| REPLY          | Allows a request to continue processing.                                |
| RESTART EXPORT | Restarts a suspended export operation.                                  |
| SUSPEND EXPORT | Suspends a running export operation.                                    |



## CANCEL REQUEST (Cancel one or more mount requests)

Use this 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 in the mount request message and can also be shown by using the QUERY REQUEST command.

### Privilege class

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

### Syntax



### Parameters

*request\_number*  
Specifies the request number of the mount request to cancel.

**ALL**  
Specifies to cancel all pending mount requests.

**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.

### Example: Cancel a mount request

Cancel request number 2.  
cancel request 2

### Related commands

Table 39. Commands related to CANCEL REQUEST

| Command       | Description  |
|---------------|--|
| QUERY REQUEST | Displays information about all pending mount requests. |
| UPDATE VOLUME | Updates the attributes of storage pool volumes.        |

# CANCEL RESTORE

## CANCEL RESTORE (Cancel a restartable restore session)

Use this command to cancel a restartable restore session. You can cancel restore sessions in the active or restartable state. Any outstanding mount requests related to this session are automatically cancelled.

To display restartable restore sessions, use the QUERY RESTORE command.

### Privilege class

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

### Syntax



### Parameters

*session\_number*  
Specifies the number for the restartable restore session. An active session is a positive number, and a restartable session is a negative number.

**ALL**  
Specifies that all the restartable restore sessions are to be cancelled.

### Example: Cancel restore operations

Cancel all restore operations.  
`cancel restore all`

### Related commands

Table 40. Commands related to CANCEL RESTORE

| Command       | Description  |
|---------------|--|
| QUERY RESTORE | Displays information about restartable restore sessions. |

## CANCEL SESSION (Cancel one or more client sessions)

Use this command to cancel existing administrative or client node sessions, and to force an administrative or client node session off the server. Any outstanding mount requests related to this session are automatically cancelled. The client node must start a new session to resume activities.

If you cancel a session that is in the idle wait (IdleW) state, the client session is automatically reconnected to the server when it starts to send data again.

If this command interrupts a process, such as backup or archive, the results of any processing active at the time of interruption are rolled back and not committed to the database.

### Privilege class

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

### Syntax

```

▶▶ CANCEL SESSION [session_number]
                  [ALL]
▶▶

```

### Parameters

*session\_number*

Specifies the number of the administrative or client node session that you want to cancel.

**ALL**

Specifies that all client node sessions are cancelled. You cannot use this parameter to cancel administrative client sessions.

### Example: Cancel a specific client node session

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

```
cancel session 3
```

### Related commands

Table 41. Commands related to CANCEL SESSION

| Command          | Description  |
|------------------|--|
| DISABLE SESSIONS | Prevents new sessions from accessing IBM Tivoli Storage Manager but permits existing sessions to continue. |
| LOCK ADMIN       | Prevents an administrator from accessing IBM Tivoli Storage Manager.                                       |
| LOCK NODE        | Prevents a client from accessing the server.   |
| QUERY SESSION    | Displays information about all active administrator and client sessions with IBM Tivoli Storage Manager.   |

## CHECKIN LIBVOLUME (Check a storage volume into a library)

Use this command to add a sequential access storage volume or a cleaning tape to the server inventory for an automated library. The server cannot use a volume that physically resides in an automated library until that volume has been checked in.

### Important:

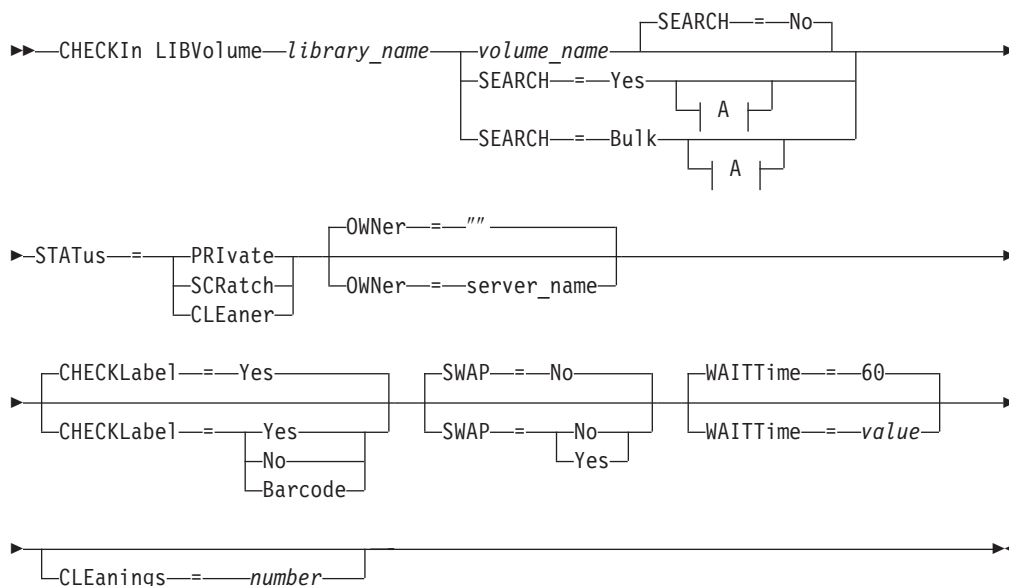
1. The CHECKIN LIBVOLUME command processing does not wait for a drive to become available, even if the drive is only in the IDLE state. If necessary, you can make a library drive available issuing the DISMOUNT VOLUME command to dismount the volume. After a library drive is available, reissue the CHECKIN LIBVOLUME command.
2. You do not define the drives, check in media, or label the volumes in an external library. The server provides an interface that external media management systems use to operate with the server. For more information, refer to the *Administrator's Guide*.
3. z/OS LAN-free volumes do not need to be checked in. These volumes are already owned by MVS™. Open system scratch tape mounts are not supported for LAN-free on z/OS.
4. When checking in WORM tapes other than 3592, you must use CHECKLABEL=YES or they will be checked in as normal read-write tapes.

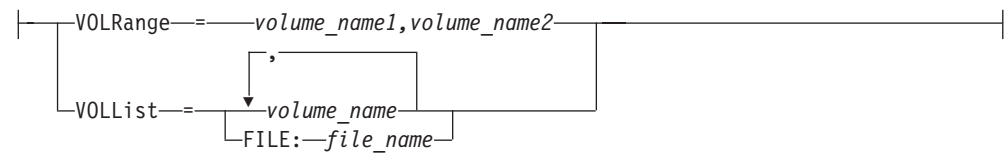
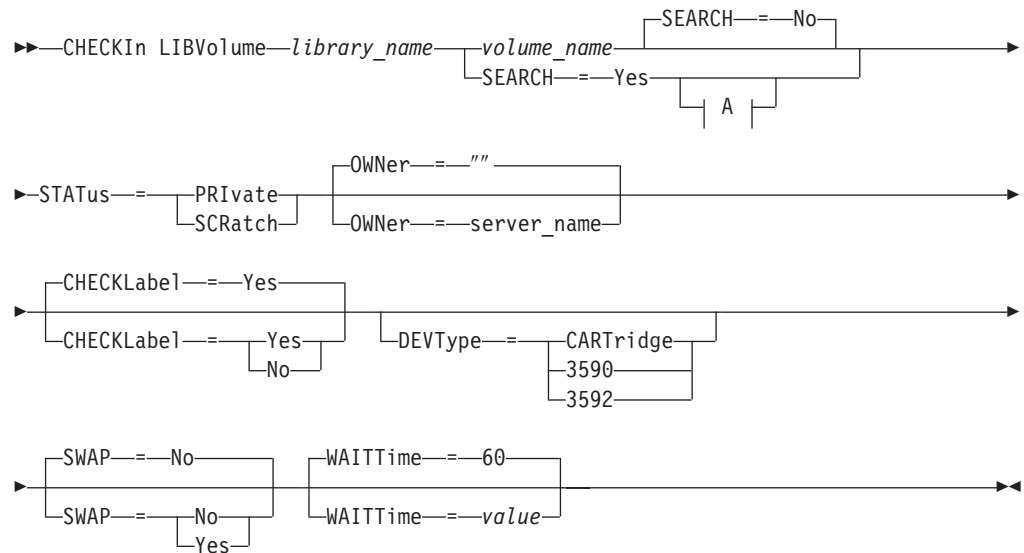
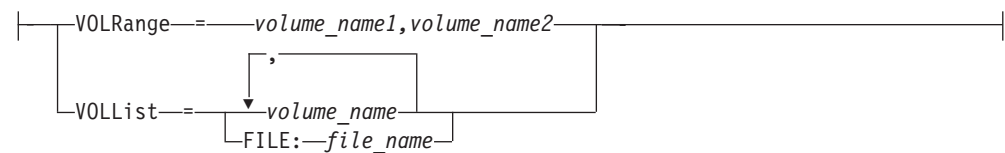
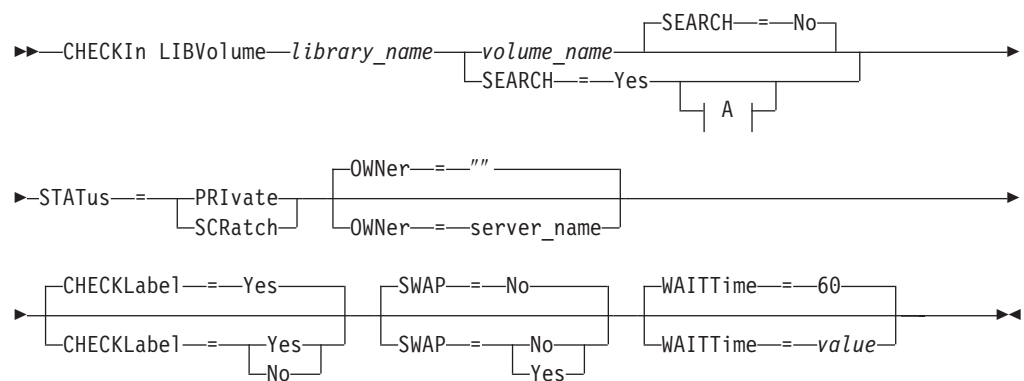
This command creates a background process that you can cancel with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

### Privilege class

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

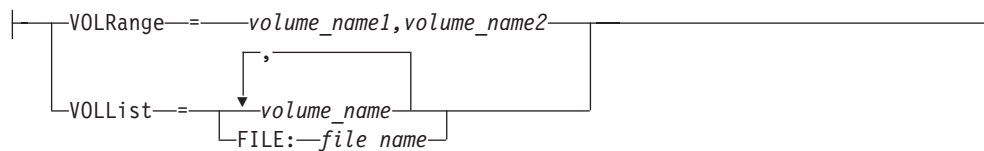
### Syntax for SCSI libraries



**A (SEARCH=Yes, SEARCH=Bulk):****Syntax for 349X libraries****A (SEARCH=Yes):****Syntax for ACSLS libraries**

## CHECKIN LIBVOLUME

### A (SEARCH=Yes):



### Parameters

#### *library\_name* (Required)

Specifies the name of the library.

#### *volume\_name*

Specifies the volume name of the storage volume being checked in. This parameter is required if SEARCH equals NO. Do not enter this parameter if the SEARCH parameter equals YES or BULK. If you are checking a volume into a SCSI library with multiple entry/exit ports, the volume in the lowest numbered slot will be checked in.

#### STATUS (Required)

Specifies the volume status. Possible values are:

##### PRivate

Specifies that the volume is a private volume that is mounted only when it is requested by name.

##### SCRatch

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

If a volume has an entry in volume history, you cannot check it in as a scratch volume.

##### CLEaner

Specifies that the volume is a cleaner cartridge and not a data cartridge. The CLEANINGS parameter is required for a cleaner cartridge and must be set to the number of cleaner uses.

CHECKLABEL=YES is not valid for checking in a cleaner cartridge. Use STATUS=CLEANER to check in a cleaner cartridge separately from a data cartridge.

#### OWNer

Specifies which library client owns a private volume in a library that is shared across a SAN. The volume for which you specify ownership must be a private volume. You cannot specify ownership for a scratch volume. Furthermore, you cannot specify an owner when you use SEARCH=YES or SEARCH=BULK.

When you issue the CHECKIN LIBVOLUME command, the Tivoli Storage Manager server validates the owner. If you did not specify this parameter, then the server uses the default and delegates volume ownership to the owning library client, as recorded in the volume history file on the library manager. If the volume is not owned by any library client, then the server delegates ownership to the library manager.

#### SEARCH

Specifies whether the server searches the library to find volumes that were not checked in. This parameter is optional. The default is NO.

Possible values are:

### No

Specifies that only the named volume is checked into the library.

**For SCSI libraries:** The server issues a request to have the volume inserted into a cartridge slot in the library or, if available, into an entry port. The cartridge slot or entry port is identified by its element address. **For 349X libraries:** The volume could already be in the library, or you could put it into the I/O station when prompted.

### Yes

Specifies that the server searches the library for volumes to be checked in. You can use the VOLRANGE or VOLLIST parameter to limit the search. When using this parameter, consider the following restrictions:

- If the library is shared between applications, the server could examine a volume required by another application. For 349X libraries, the server queries the library manager to determine all volumes that are assigned to the SCRATCH or PRIVATE category and to the INSERT category.
- For SCSI libraries, do not specify both SEARCH=YES and CHECKLABEL=NO in the same command.

### Bulk

Specifies that the server searches the library's entry/exit ports for volumes that can be checked in automatically. This option only applies to SCSI libraries.

### Important:

1. Do not specify both CHECKLABEL=NO and SEARCH=BULK.
2. You can use the VOLRANGE or VOLLIST parameter to limit the search.

### VOLRange

Specifies a range of volume names separated by commas. You can use this parameter to limit the search for volumes to be checked in when you specify SEARCH=YES (349X, ACSLS, and SCSI libraries) or SEARCH=BULK (SCSI libraries only). If there are no volumes in the library that are within the specified range, the command completes without errors.

Specify only volume names that can be numerically incremented. In addition to the incremental area, a volume name can include an alphanumeric prefix and an alphanumeric suffix, for example:

| Parameter              | Description  |
|------------------------|--|
| volrange=bar110,bar130 | The 21 volumes are checked in: bar110, bar111, bar112,...bar129, bar130. |
| volrange=bar11a,bar13a | The 3 volumes are checked in: bar11a, bar12a, bar13a.                    |
| volrange=123400,123410 | The 11 volumes are checked in: 123400, 123401, ...123409, 123410.        |

### VOLList

Specifies a list of volumes. You can use this parameter to limit the search for volumes to be checked in when you specify SEARCH=YES (349X, ACSLS, and SCSI libraries) or SEARCH=BULK (SCSI libraries only). If there are no volumes in the library that are in the list, the command completes without errors.

Possible values are:

## CHECKIN LIBVOLUME

### *volume\_name*

Specifies one or more volumes names that are separated by commas and no intervening spaces. For example: VOLLIST=TAPE01,TAPE02.

### **FILE:** *file\_name*

Specifies the name of a file that contains a list of volumes for the command. In the file, each volume name must be on a separate line. Blank lines and comment lines that begin with an asterisk are ignored. For example, to use volumes TAPE01, TAPE02 and TAPE03, create a file, TAPEVOL, that contains these lines:

```
TAPE01
TAPE02
TAPE03
```

You can specify the volumes for the command as follows:  
VOLLIST=FILE:TAPEVOL.

**Attention:** The file name is case-sensitive.

### **CHECKLabel**

Specifies how or whether the server should read sequential media labels of volumes. This parameter is optional. The default is YES.

Possible values are:

#### **Yes**

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

#### **Attention:**

1. For optical volumes (write-once and rewritable), you must specify YES.
2. For SCSI libraries, do not specify both SEARCH=YES and CHECKLABEL=NO in the same command.
3. For WORM media other than 3592, you must specify YES.

#### **No**

Specifies that the media label is not read during check-in. However, suppressing label checking can result in future errors (for example, either a wrong label or an improperly labeled volume can cause an error). For 349X and ACSLS libraries, specify NO to avoid loading cartridges into a drive to read the media label. These libraries always return the external label information on cartridges, and Tivoli Storage Manager uses that information.

### **Barcode**

Specifies that the server reads the bar code label if the library has a bar code reader and the volumes have external bar code labels. You can decrease the check-in time by using the bar code. This parameter applies only to SCSI libraries.

If the bar code reader cannot read the bar code label, or if the tape does not have a bar code label, the server mounts the tape and reads the internal label.

### **DEVType**

Specifies the device type for the volume being checked in. This parameter is required if none of the drives in this library have defined paths.

### **CARtridge**

Specifies that the device type for the volume being checked in is 3490.



**3590**

Specifies that the device type for the volume being checked in is 3590.

**3592**

Specifies that the device type for the volume being checked in is 3592.

**SWAP**

Specifies whether the server swaps volumes if an empty library slot is not available. The volume selected for the swap operation (target swap volume) is ejected from the library and replaced with the volume being checked in. The server identifies a target swap volume by checking for an available scratch volume. If none exists, the server identifies the least frequently mounted volume.

This parameter is optional. The default is NO. This parameter only applies if there is a volume name specified in the command. Possible values are:

**No**

Specifies that the server checks in the volume only if an empty slot is available.

**Yes**

Specifies that if an empty slot is not available, the server swaps cartridges to check in the volume.

**WAITTime**

Specifies the number of minutes that the server waits for you to reply or respond to a request. Specify a value in the range 0-9999. If you want to be prompted by the server, specify a wait time greater than zero. The default value is 60 minutes. For example, suppose the server prompts you to insert a tape into the entry/exit port of a library. If you specified a wait time of 60 minutes, the server issues a request and waits 60 minutes for you to reply. Suppose, on the other hand, you specify a wait time of 0. If you have already inserted a tape, a wait time of zero causes the operation to continue without prompting. If you have *not* inserted a tape, a wait time of zero will cause the operation to fail.

**CLEAnings**

Enter the recommended value for the individual cleaner cartridge (usually indicated on the cartridge). Cleanings apply only to SCSI libraries. This parameter is required if STATUS=CLEANER.

If more than one cleaner is checked into the library, only one is used until its CLEANINGS value decreases to zero. Another cleaner is then be selected, and the first cleaner can be checked out and discarded.

**Example: Check a volume into a SCSI library**

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

```
checkin libvolume auto wpdv00 status=scratch
```

**Example: Use a bar code reader to scan a library for a cleaner cartridge**

Scan a SCSI library named AUTOLIB1 and, using the bar code reader, look for cleaner cartridge CLNV. Use SEARCH=YES, but limit the search by using the VOLLIST parameter.

```
checkin libvolume autolib1 search=yes vollist=cleanv status=cleaner  
cleanings=10 checklabel=barcode
```

## CHECKIN LIBVOLUME

### Example: Scan a library to put unused volumes in a specific range in scratch status

Scan a 349X library named ABC, and limit the search to a range of unused volumes BAR110 to BAR130 and put them in scratch status.

```
checkin libvolume abc search=yes volrange=bar110,bar130
status=scratch
```

### Example: Scan a library to put a specific volume in scratch status

Use the barcode reader to scan a SCSI library named MYLIB for VOL1, and put it in scratch status.

```
checkin libvolume mylib search=yes vollist=vol1 status=scratch
checklabel=barcode
```

## Related commands

Table 42. Commands related to CHECKIN LIBVOLUME

| Command            | Description  |
|--------------------|--|
| AUDIT LIBRARY      | Ensures that an automated library is in a consistent state.              |
| CANCEL PROCESS     | Cancels a background server process.                                     |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library.                     |
| DEFINE LIBRARY     | Defines an automated or manual library.                                  |
| DEFINE VOLUME      | Assigns a volume to be used for storage within a specified storage pool. |
| DISMOUNT VOLUME    | Dismounts a sequential, removable volume by the volume name.             |
| LABEL LIBVOLUME    | Labels volumes in manual or automated libraries.                         |
| QUERY LIBRARY      | Displays information about one or more libraries.                        |
| QUERY LIBVOLUME    | Displays information about a library volume.                             |
| QUERY PROCESS      | Displays information about background processes.                         |
| REPLY              | Allows a request to continue processing.                                 |
| UPDATE LIBVOLUME   | Changes the status of a storage volume.                                  |

## CHECKOUT LIBVOLUME (Check a storage volume out of a library)

Use this command to remove a sequential access storage volume from the server inventory for an automated library. 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.

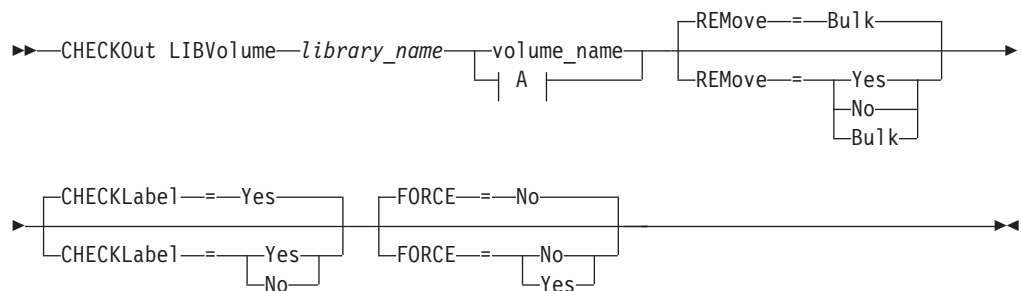
### Restrictions:

1. Check out processing does not wait for a drive to become available, even if the drive is in the IDLE state. If necessary, you can make a library drive available by dismounting the volume with the DISMOUNT VOLUME command. After a drive is available, the CHECKOUT LIBVOLUME command can be reissued.
2. Before checking out volumes from a 349X library, ensure that the 349x Cartridge Input and Output facility has enough empty slots for the volumes to be checked out. The 3494 Library Manager does not inform an application that the Cartridge Input and Output facility is full. It accepts requests to eject a cartridge and waits until the Cartridge Input and Output facility is emptied before returning to the server. Tivoli Storage Manager may appear to be hung when it is not. You should check the library and clear any intervention requests.
3. Before checking volumes out of an ACSLS library, ensure that the CAP priority in ACSLS is greater than zero. If the CAP priority is zero, then you must specify a value for the CAP parameter on the CHECKOUT LIBVOLUME command.

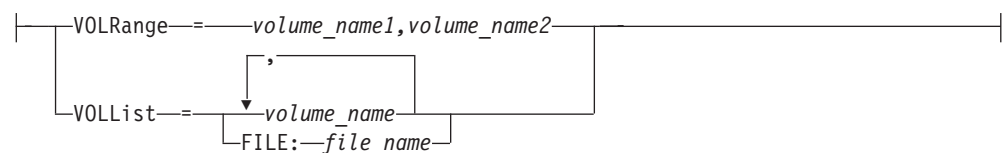
### Privilege class

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

### Syntax for SCSI library

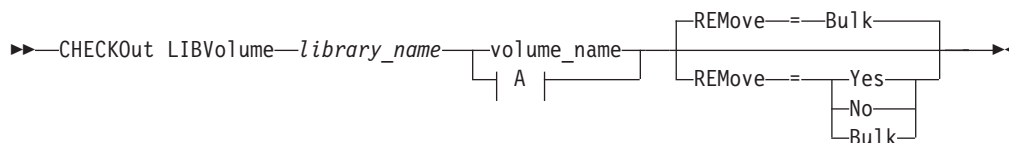


**A :**

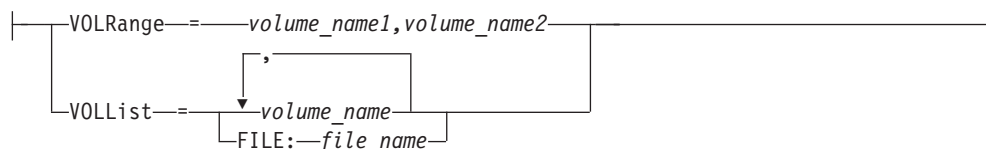


## CHECKOUT LIBVOLUME

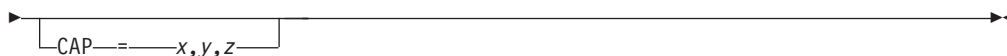
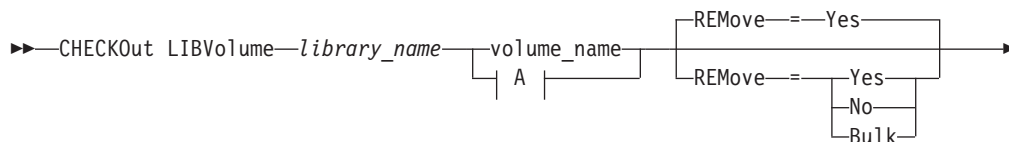
### Syntax for 349X library



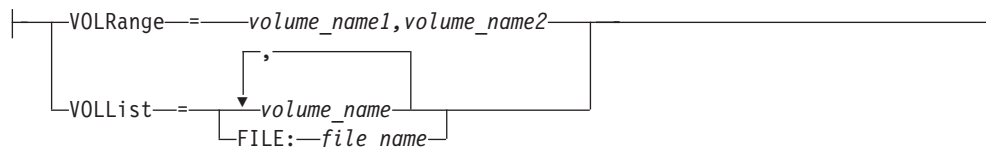
**A :**



### Syntax for ACSLS library



**A :**



## Parameters

### *library\_name* (Required)

Specifies the name of the library.

### *volume\_name*

Specifies the volume name.

### **VOLRange**

Specifies two volume names separated by a comma. This parameter is a range of volumes to be checked out. If there are no volumes in the library that are within the specified range, the command completes without errors.

Specify only volume names that can be numerically incremented. In addition to the incremental area, a volume name can include an alphanumeric prefix and an alphanumeric suffix, for example:

| Parameter              | Description   |
|------------------------|---|
| volrange=bar110,bar130 | The 21 volumes are checked out: bar110, bar111, bar112,...bar129, bar130. |
| volrange=bar11a,bar13a | The 3 volumes are checked out: bar11a, bar12a, bar13a.                    |
| volrange=123400,123410 | The 11 volumes are checked out: 123400, 123401, ...123409, 123410.        |

**VOLLIST**

Specifies a list of volumes to check out. If there are no volumes in the library that are in the list, the command completes without errors.

Possible values are:

*volume\_name*

Specifies the names of one or more values used for the command.

Example: VOLLIST=TAPE01,TAPE02.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes for the command. In the file, each volume name must be on a separate line. Blank lines and comment lines that begin with an asterisk are ignored. For example, to use volumes TAPE01, TAPE02 and TAPE03, create a file, TAPEVOL, that contains these lines:

```
TAPE01
TAPE02
TAPE03
```

You can specify the volumes for the command as follows:

VOLLIST=FILE:TAPEVOL.

**Attention:** The file name is case-sensitive.

**REMOve**

Specifies that the server tries to move the volume out of the library and into the convenience I/O station or entry/exit ports. This parameter is optional. Possible values are YES, BULK, and NO. Possible values, depending on the type of library, are YES, BULK, and NO. The response of the server to each of those options and the default values are described in the following sections.

**349X libraries:** The default is BULK. The following table shows how the server responds for 349X libraries.

Table 43. How the Tivoli Storage Manager Server Responds for 349X Libraries

| REMOVE=YES  | REMOVE=BULK   | REMOVE=NO   |
|---|---|---|
| The 3494 Library Manager ejects the cartridge to the convenience I/O station. | The 3494 Library Manager ejects the cartridge to the high-capacity output facility. | The 3494 Library Manager does not eject the volume.<br><br>The server leaves the cartridge in the library in the INSERT category for use by other applications. |

**SCSI libraries:** The default is BULK. The following table shows how the server responds for a SCSI libraries.

## CHECKOUT LIBVOLUME

Table 44. How the Tivoli Storage Manager Server Responds for SCSI Libraries

| If a library . . .  | And REMOVE=YES,<br>then...   | And REMOVE=BULK,<br>then...  | And REMOVE=NO,<br>then...   |
|---|--|--|---|
| <i>Does not have entry/exit ports</i>                           | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server then prompts you to remove the cartridge from the slot and to issue a REPLY command. | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a eREPLY command. | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a REPLY command. |
| <i>Has entry/exit ports and an entry/exit port is available</i> | The server moves the cartridge to the available entry/exit port and specifies the port address in a message.<br><br>The server then prompts you to remove the cartridge from the slot and to issue a REPLY command.        | The server moves the cartridge to the available entry/exit port and specifies the port address in a message.<br><br>The server does not prompt you to remove the cartridge and does not request a REPLY command.         | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a REPLY command. |
| <i>Has entry/exit ports, but no ports are available</i>         | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server then prompts you to remove the cartridge from the slot and to issue a REPLY command. | The server waits for an entry/exit port to be made available.  | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a REPLY command. |

**ACSLs libraries:** The default is YES. The following table shows how the server responds for ACSLS libraries.

Table 45. How the Tivoli Storage Manager Server Responds for ACSLS Libraries

| REMOVE=YES or REMOVE=BULK   | REMOVE=NO  |
|---|--|
| The server ejects the cartridge to the convenience I/O station, and deletes the volume entry from the server library inventory. | The server does not eject the cartridge. The server deletes the volume entry from the server library inventory and leaves the volume in the library. |

### CHECKLabel

Specifies how or whether the server reads sequential media labels of volumes.

**Attention:** This parameter does not apply to IBM 349X or ACSLS libraries.

This parameter is optional. The default is YES. Possible values are:

#### Yes

Specifies that the server attempts to read the media label to verify that the correct volume is being checked out.

**No**

Specifies that during checkout the media label is not read. This improves performance because the read process does not occur.

**FORCE**

Specifies whether the server checks out a volume if an input/output (I/O) error occurs when reading the label.

**Attention:** This parameter does not apply to IBM 349X or ACSLS libraries.

This parameter is optional. The default is NO. Possible values are:

**No**

The server does not check out a storage volume if an I/O error occurs when reading the label.

**Yes**

The server checks out the storage volume even if an I/O error occurs.

**CAP**

Specifies which cartridge access port (CAP) to use for ejecting volumes if you specify REMOVE=YES. This parameter applies to volumes in ACSLS libraries only. If a CAP priority greater than zero does not exist in the library, this parameter is required. If a CAP priority greater than zero does exist in the library, this parameter is optional. If you do not specify the parameter, the ACSLS server will choose any available CAP with a priority greater than zero.

To display valid CAP identifiers ( $x,y,z$ ), issue the QUERY CAP command with ALL specified from the Automated Cartridge System System Administrator (ACSSA) console on the ACSLS server host. The identifiers are as follows:

- $x$  The Automated Cartridge System (ACS) ID. This identifier can be a number between 0-126.
- $y$  The Library Storage Module (LSM) ID. This identifier can be a number between 0-23.
- $z$  The CAP ID. This identifier can be a number between 0-11.

For more information, refer to your StorageTek documentation.

**Example: Check out a volume and check the label**

Check out the volume named EXB004 from the library named FOREST. Read the label to verify the volume name, but do not move the volume out of the library.

```
checkout libvolume forest exb004 checklabel=yes remove=no
```

**Related commands**

Table 46. Commands related to CHECKOUT LIBVOLUME

| Command           | Description  |
|-------------------|--|
| AUDIT LIBRARY     | Ensures that an automated library is in a consistent state.              |
| CANCEL PROCESS    | Cancels a background server process.                                     |
| CHECKIN LIBVOLUME | Checks a storage volume into an automated library.                       |
| DEFINE LIBRARY    | Defines an automated or manual library.                                  |
| DEFINE VOLUME     | Assigns a volume to be used for storage within a specified storage pool. |

## CHECKOUT LIBVOLUME

*Table 46. Commands related to CHECKOUT LIBVOLUME (continued)*

| Command          | Description                                       |
|------------------|---|
| LABEL LIBVOLUME  | Labels volumes in manual or automated libraries.  |
| QUERY LIBRARY    | Displays information about one or more libraries. |
| QUERY LIBVOLUME  | Displays information about a library volume.      |
| QUERY PROCESS    | Displays information about background processes.  |
| REPLY            | Allows a request to continue processing.          |
| UPDATE LIBVOLUME | Changes the status of a storage volume.           |



## CLEAN DRIVE (Clean a drive)

Use this command when you want Tivoli Storage Manager to immediately load a cleaner cartridge into a drive regardless of the cleaning frequency.

There are special considerations if you plan to use this command with a SCSI library that provides automatic drive cleaning through its device hardware. See the *Administrator's Guide* for details.

**Restriction:** You cannot run the CLEAN DRIVE command for a drive whose only path source is a NAS file server.

### Privilege class

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

### Syntax

►►—CLEAN DRIVE—*library\_name*—*drive\_name*—►►

### Parameters

*library\_name* **(Required)**

Specifies the name of the library to which the drive is assigned.

*drive\_name* **(Required)**

Specifies the name of the drive.

### Example: Clean a specific tape drive

You have already defined a library named AUTOLIB by using the DEFINE LIBRARY command, and you have already checked a cleaner cartridge into the library using the CHECKIN LIBVOL command. Inform the server that TAPEDRIVE3 in this library requires cleaning.

```
clean drive autolib tapedrive3
```

### Related commands

Table 47. Commands related to CLEAN DRIVE

| Command            | Description  |
|--------------------|--|
| CHECKIN LIBVOLUME  | Checks a storage volume into an automated library.   |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library. |
| DEFINE DRIVE       | Assigns a drive to a library.                        |
| DEFINE LIBRARY     | Defines an automated or manual library.              |
| DELETE DRIVE       | Deletes a drive from a library.                      |
| QUERY DRIVE        | Displays information about drives.                   |
| UPDATE DRIVE       | Changes the attributes of a drive.                   |

## COMMIT (Control committing of commands in a macro)

Use this command to control when a command is committed in a macro and to update the database when commands complete processing. When issued from the console mode of the administrative client, this command does not generate a message.

If an error occurs while processing the commands in a macro, the server stops processing the macro and rolls back any changes (since the last COMMIT). After a command is committed, it cannot be rolled back.

Ensure that your administrative client session is not running with the ITEMCOMMIT option if you want to control command processing. The ITEMCOMMIT option commits commands inside a script or a macro as *each* command is processed.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—COMMIT—◄◄

### Parameters

None.

### Example: Control committing of commands in a macro

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 is 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 */
```

#### Command

```
macro reg.adm
```

### Related commands

Table 48. Commands related to COMMIT

| Command  | Description  |
|----------|--|
| MACRO    | Runs a specified macro file.   |
| ROLLBACK | Discards any uncommitted changes to the database since the last COMMIT was executed. |

---

## COPY commands

Use the COPY commands to create a copy of Tivoli Storage Manager objects.

The following is a list of COPY commands for Tivoli Storage Manager:

- “COPY ACTIVATEDATA (Copy active backup data from a primary storage pool to an active-data pool)” on page 86
- “COPY CLOPTSET (Copy a client option set)” on page 90
- “COPY DOMAIN (Copy a policy domain)” on page 91
- “COPY MGMTCLASS (Copy a management class)” on page 93
- “COPY POLICYSET (Copy a policy set)” on page 95
- “COPY PROFILE (Copy a profile)” on page 97
- “COPY SCHEDULE (Copy a client or an administrative command schedule)” on page 99
- “COPY SCRIPT (Copy a Tivoli Storage Manager script)” on page 103
- “COPY SERVERGROUP (Copy a server group)” on page 104

## COPY ACTIVEDATA (Copy active backup data from a primary storage pool to an active-data pool)

Use this command to copy active versions of backup data from a primary storage pool to an active-data pool. The primary benefit of active-data pools is fast client restores. Copy your active data regularly to ensure that the data is protected in case of a disaster.

If a file already exists in the active-data pool, the file is not copied unless the copy of the file in the active-data pool is marked damaged. However, a new copy is not created if the file in the primary storage pool is also marked damaged. In a random-access storage pool, neither cached copies of migrated files nor damaged primary files are copied.

If migration for a storage pool starts while active data is being copied, some files might be migrated before they are copied. For this reason, you should copy active data from storage pools that are higher in the migration hierarchy before copying active data from storage pools that are lower. Be sure a copy process is complete before beginning another.

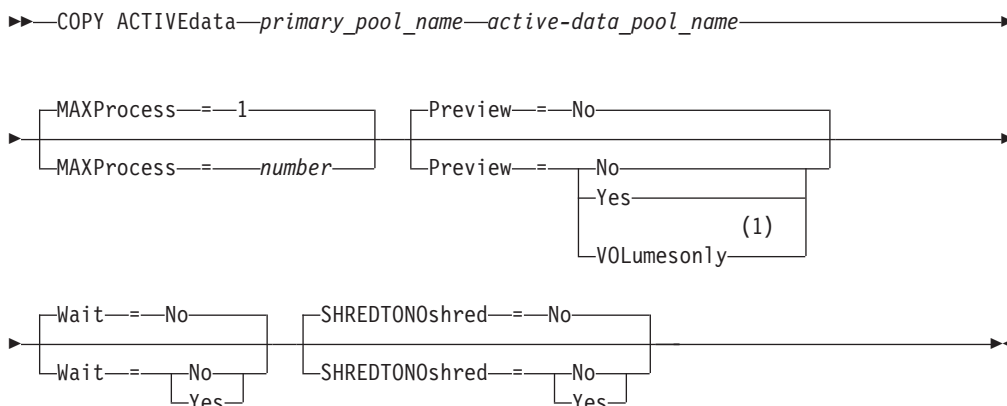
### Remember:

- You can only copy active data from storage pools that have a data format of NATIVE or NONBLOCK.
- Issuing this command for a primary storage pool that is set up for deduplication removes duplicate data, if the active-data pool is also set up for deduplication.

### Privilege class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the active-data pool from which active versions of backup data are being copied.

### Syntax



### Notes:

- 1 The **VOLUMESONLY** parameter applies to sequential-access storage pools only.

## Parameters

### *primary\_pool\_name* (Required)

Specifies the primary storage pool.

### *active\_data\_pool\_name* (Required)

Specifies the active-data pool.

### MAXProcess

Specifies the maximum number of parallel processes to use for copying files. This parameter is optional. Enter a value from 1 to 999. The default is 1.

Using multiple, parallel processes may improve throughput for the COPY ACTIVATEDATA command. The expectation is that the time needed to copy active data will be decreased by using multiple processes. However, when multiple processes are running, in some cases one or more of the processes might need to wait to use a volume that is already in use by a different COPY ACTIVATEDATA process.

When determining this value, consider the number of logical and physical drives that can be dedicated to this operation. To access a sequential-access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential-access storage pools that are involved when copying active data.

Each process needs a mount point for active-data pool volumes, and, if the device type is not FILE, each process also needs a drive. If you are copying active data from a sequential-access storage pool, each process needs an additional mount point for primary storage pool volumes and, if the device type is not FILE, an additional drive. For example, suppose you specify a maximum of 3 processes to copy a primary sequential storage pool to an active-data pool of the same device class. Each process requires two mount points and two drives. To run all three processes, the device class must have a mount limit of at least six, and at least six mount points and six drives must be available.

To use the **PREVIEW** parameter, only one process is used, and no mount points or drives are needed.

### Preview

Specifies whether you want to preview but not actually copy any active data. The preview displays the number of files and bytes to be copied and a list of the primary storage pool volumes that you must mount. This parameter is optional. The default is NO. Possible values are:

#### No

Specifies that active data will be copied.

#### Yes

Specifies that you want to preview the process but not copy any data.

### VOLUMesonly

Specifies that you want to preview the process only as a list of the volumes that must be mounted. This choice requires the least processing time.

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is NO. Possible values are:

## COPY ACTIVE DATA

### No

Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed. Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files may have already been copied prior to the cancellation.

### Yes

Specifies that the server performs this operation in the foreground. You must wait for the operation to complete before continuing with other tasks. The server displays the output messages to the administrative client when the operation completes.

You cannot specify WAIT=YES from the server console.

## SHREDTONOshred

Specifies whether data should be copied from a primary storage pool that enforces shredding to an active-data pool that does not enforce shredding. This parameter is optional. The default value is NO. Possible values are:

### No

Specifies that the server does not allow data to be copied from a primary storage pool that enforces shredding to an active-data pool that does not enforce shredding. If the primary storage pool enforces shredding and the active-data pool does not, the operation will fail.

### Yes

Specifies that the server does allow data to be copied from a primary storage pool that enforces shredding to an active-data pool that does not enforce shredding. The data in the active-data pool will not be shredded when it is deleted.

## Example: Copy primary storage pool data to active-data pool

Copy the active data from a primary storage pool named PRIMARY\_POOL to the active-data pool named ACTIVEPOOL. Issue the command:

```
copy activedata primary_pool activepool
```

## Related commands

Table 49. Commands related to COPY ACTIVE DATA

| Command        | Description   |
|----------------|---|
| DEFINE DOMAIN  | Defines a policy domain that clients can be assigned to.              |
| DEFINE STGPOOL | Defines a storage pool as a named collection of server storage media. |
| EXPORT NODE    | Copies client node information to external media.                     |
| EXPORT SERVER  | Copies all or part of the server to external media.                   |
| IMPORT NODE    | Restores client node information from external media.                 |

Table 49. Commands related to COPY ACTIVATEDATA (continued)

| Command         | Description   |
|-----------------|---|
| IMPORT SERVER   | Restores all or part of the server from external media.                                       |
| MOVE NODEDATA   | Moves data for one or more nodes, or a single node with selected file spaces.                 |
| QUERY CONTENT   | Displays information about files in a storage pool volume.                                    |
| QUERY DOMAIN    | Displays information about policy domains.  |
| QUERY NODE      | Displays partial or complete information about one or more clients.                           |
| QUERY NODEDATA  | Displays information about the location and size of data for a client node.                   |
| QUERY STGPOOL   | Displays information about storage pools.   |
| RESTORE STGPOOL | Restores files to a primary storage pool from copy storage pools.                             |
| RESTORE VOLUME  | Restores files stored on specified volumes in a primary storage pool from copy storage pools. |
| UPDATE DOMAIN   | Changes the attributes of a policy domain.  |
| UPDATE STGPOOL  | Changes the attributes of a storage pool.   |

## COPY CLOPTSET (Copy a client option set)

Use this command to copy a client option 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 client node is assigned.

### Syntax

►►—COPY CLOptset—*current\_option\_set\_name*—*new\_option\_set\_name*—►►

### Parameters

*current\_option\_set\_name* **(Required)**

Specifies the name of the client option set to be copied.

*new\_option\_set\_name* **(Required)**

Specifies the name of the new client option set. The maximum length of the name is 64 characters.

### Example: Copy a client option set

Copy a client option set named ENG to a new client option set named ENG2.

```
copy cloptset eng eng2
```

### Related commands

Table 50. Commands related to COPY CLOPTSET

| Command          | Description  |
|------------------|--|
| DEFINE CLIENTOPT | Adds a client option to a client option set.                           |
| DEFINE CLOPTSET  | Defines a client option set.   |
| DELETE CLIENTOPT | Deletes a client option from a client option set.                      |
| DELETE CLOPTSET  | Deletes a client option set.   |
| QUERY CLOPTSET   | Displays information about a client option set.                        |
| UPDATE CLIENTOPT | Updates the sequence number of a client option in a client option set. |
| UPDATE CLOPTSET  | Updates the description of a client option set.                        |



## COPY DOMAIN (Copy a policy domain)

Use this command to create a copy of a policy domain.

The server copies the following information to the new domain:

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

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► `COPY DOMAIN` *current\_domain\_name* *new\_domain\_name* ◀◀

### Parameters

*current\_domain\_name* **(Required)**

Specifies the policy domain to copy.

*new\_domain\_name* **(Required)**

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

### Example: Copy a policy domain to a new policy domain

Copy the policy domain PROG1 to new policy domain PROG2.

```
copy domain prog1 prog2
```

### Related commands

Table 51. Commands related to COPY DOMAIN

| Command            | Description  |
|--------------------|--|
| ACTIVATE POLICYSET | Validates and activates a policy set.  |
| COPY MGMTCLASS     | Creates a copy of a management class.  |
| DEFINE COPYGROUP   | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE DOMAIN      | Defines a policy domain that clients can be assigned to.                                   |
| DEFINE MGMTCLASS   | Defines a management class.  |
| DEFINE POLICYSET   | Defines a policy set within the specified policy domain.                                   |
| DELETE COPYGROUP   | Deletes a backup or archive copy group from a policy domain and policy set.                |
| DELETE DOMAIN      | Deletes a policy domain along with any policy objects in the policy domain.                |

## COPY DOMAIN

Table 51. Commands related to COPY DOMAIN (continued)

| Command            | Description  |
|--------------------|--|
| DELETE MGMTCLASS   | Deletes a management class and its copy groups from a policy domain and policy set.                  |
| QUERY COPYGROUP    | Displays the attributes of a copy group.   |
| QUERY DOMAIN       | Displays information about policy domains.   |
| QUERY MGMTCLASS    | Displays information about management classes.   |
| QUERY POLICYSET    | Displays information about policy sets.  |
| REGISTER NODE      | Defines a client to the server and sets options for that user.                                       |
| UPDATE COPYGROUP   | Changes one or more attributes of a copy group.  |
| UPDATE DOMAIN      | Changes the attributes of a policy domain.   |
| UPDATE MGMTCLASS   | Changes the attributes of a management class.  |
| UPDATE POLICYSET   | Changes the description of a policy set.   |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

## COPY MGMTCLASS (Copy a management class)

Use this command to create a copy of a management class within the same policy set.

The server copies the following information to the new management class:

- Management class description
- Copy groups defined to the management class
- Any attributes for managing files for Tivoli Storage Manager for Space Management clients

### 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—domain_name—policy_set_name—————►
►—current_class_name—new_class_name—————►►
```

### Parameters

*domain\_name* **(Required)**

Specifies the policy domain to which the management class belongs.

*policy\_set\_name* **(Required)**

Specifies the policy set to which the management class belongs.

*current\_class\_name* **(Required)**

Specifies the management class to copy.

*new\_class\_name* **(Required)**

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

### Example: Copy a management class to a new management class

Copy the management class ACTIVEFILES to a new management class, FILEHISTORY. The management class is in policy set VACATION in the EMPLOYEE\_RECORDS policy domain.

```
copy mgmtclass employee_records vacation
activefiles filehistory
```

### Related commands

Table 52. Commands related to COPY MGMTCLASS

| Command          | Description  |
|------------------|--|
| DEFINE COPYGROUP | Defines a copy group for backup or archive processing within a specified management class. |
| DELETE MGMTCLASS | Deletes a management class and its copy groups from a policy domain and policy set.        |
| QUERY COPYGROUP  | Displays the attributes of a copy group.   |

## COPY MGMTCLASS

*Table 52. Commands related to COPY MGMTCLASS (continued)*

| Command          | Description                                     |
|------------------|---|
| QUERY MGMTCLASS  | Displays information about management classes.  |
| QUERY POLICYSET  | Displays information about policy sets.         |
| UPDATE COPYGROUP | Changes one or more attributes of a copy group. |
| UPDATE MGMTCLASS | Changes the attributes of a management class.   |

## COPY POLICYSET (Copy a policy set)

Use this command to copy a policy set (including the ACTIVE policy set) within the same policy domain.

The server copies the following information to the new policy set:

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

### Syntax

```
►►—COPY Policyset—domain_name—current_set_name—new_set_name—————►►
```

### Parameters

*domain\_name* **(Required)**

Specifies the policy domain to which the policy set belongs.

*current\_set\_name* **(Required)**

Specifies the policy set to copy.

*new\_set\_name* **(Required)**

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

### Example: Copy a policy set to a new policy set

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

```
copy policyset employee_records vacation holiday
```

### Related commands

Table 53. Commands related to COPY POLICYSET

| Command            | Description  |
|--------------------|--|
| ACTIVATE POLICYSET | Validates and activates a policy set.  |
| COPY MGMTCLASS     | Creates a copy of a management class.  |
| DEFINE MGMTCLASS   | Defines a management class.  |
| DELETE POLICYSET   | Deletes a policy set, including its management classes and copy groups, from a policy domain.        |
| QUERY POLICYSET    | Displays information about policy sets.  |
| UPDATE POLICYSET   | Changes the description of a policy set.   |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |



## COPY PROFILE (Copy a profile)

Use this command on a configuration manager to copy a profile and all its associated object names to a new profile.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► `COPY PROFILE` *current\_profile\_name* *new\_profile\_name* ◀◀

### Parameters

*current\_profile\_name* **(Required)**

Specifies the profile to copy.

*new\_profile\_name* **(Required)**

Specifies the name of the new profile. The maximum length of the profile name is 30 characters.

### Example: Make a copy of a profile

Copy a profile named VAL to a new profile named VAL2.

```
copy profile val val2
```

### Related commands

Table 54. Commands related to COPY PROFILE

| Command                | Description   |
|------------------------|---|
| DEFINE PROFASSOCIATION | Associates objects with a profile.  |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers.          |
| DEFINE SUBSCRIPTION    | Subscribes a managed server to a profile.                                   |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.                        |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                             |
| DELETE SUBSCRIBER      | Deletes obsolete managed server subscriptions.                              |
| DELETE SUBSCRIPTION    | Deletes a specified profile subscription.                                   |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                           |
| NOTIFY SUBSCRIBERS     | Notifies servers to refresh their configuration information.                |
| QUERY PROFILE          | Displays information about configuration profiles.                          |
| QUERY SUBSCRIBER       | Displays information about subscribers and their subscriptions to profiles. |
| QUERY SUBSCRIPTION     | Displays information about profile subscriptions.                           |

## COPY PROFILE

*Table 54. Commands related to COPY PROFILE (continued)*

| Command           | Description  |
|-------------------|--|
| SET CONFIGMANAGER | Specifies whether a server is a configuration manager.         |
| UNLOCK PROFILE    | Enables a locked profile to be distributed to managed servers. |
| UPDATE PROFILE    | Changes the description of a profile.                          |



## COPY SCHEDULE (Copy a client or an administrative command schedule)

Use this command to create a copy of a 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.

*Table 55. Commands related to COPY SCHEDULE*

| Command            | Description   |
|--------------------|---|
| DEFINE ASSOCIATION | Associates clients with a schedule.                                     |
| DEFINE SCHEDULE    | Defines a schedule for a client operation or an administrative command. |
| DELETE SCHEDULE    | Deletes a schedule from the database.                                   |
| QUERY SCHEDULE     | Displays information about schedules.                                   |
| UPDATE SCHEDULE    | Changes the attributes of a schedule.                                   |

## COPY SCHEDULE (Create a copy of a schedule for client operations)

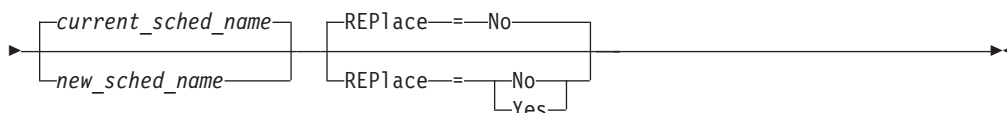
Use the COPY SCHEDULE command to create a copy of a schedule for client operations. You can copy a schedule within a policy domain or from one policy domain to another policy domain. Use the DEFINE ASSOCIATION command to associate the new schedule with the client nodes.

### Privilege class

To copy a client schedule, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which you are copying the schedule.

### Syntax

►►—COPY SCHEDULE—*current\_domain\_name*—*current\_sched\_name*—*new\_domain\_name*—►►



### Parameters

#### *current\_domain\_name* (Required)

Specifies the name of the policy domain that contains the schedule you want to copy.

#### *current\_sched\_name* (Required)

Specifies the name of the schedule you want to copy.

#### *new\_domain\_name* (Required)

Specifies the name of a policy domain to which you want to copy the new schedule.

#### *new\_sched\_name*

Specifies the name of the new schedule. You can specify up to 30 characters for the name.

If you do not specify this name, the name of the original schedule is used.

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

### REPLACE

Specifies whether to replace a client schedule. The default is NO. The values are:

#### No

Specifies that a client schedule is not replaced.

#### Yes

Specifies that a client schedule is replaced.

**Example: Copy a schedule from one policy domain to another**

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.

```
copy schedule employee_records weekly_backup  
prog1 weekly_back2
```

### COPY SCHEDULE (Create a copy of a schedule for administrative operations)

Use the COPY SCHEDULE command to create a copy of an administrative command schedule.

#### Privilege class

To copy an administrative command schedule, you must have system privilege.

#### Syntax

```
►► COPY SCHEDULE current_sched_name new_sched_name Type=Administrative►►
```



#### Parameters

##### *current\_schedule\_name* (Required)

Specifies the name of the schedule you want to copy.

##### *new\_schedule\_name* (Required)

Specifies the name of the new schedule. You can specify up to 30 characters for the name.

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

##### Type=**Administrative**

Specifies that an administrative command schedule is to be copied.

##### **REPLACE**

Specifies whether to replace an administrative command schedule. The default is NO. The values are:

##### **No**

Specifies that an administrative command schedule is not replaced.

##### **Yes**

Specifies that an administrative command schedule is replaced.

#### Example: Copy an administrative command schedule to another schedule

Copy the administrative command schedule, DATA\_BACKUP and name the schedule DATA\_ENG. If there is already a schedule with this name, replace it.

```
copy schedule data_backup data_eng
type=administrative replace=yes
```

## COPY SCRIPT (Copy a Tivoli Storage Manager script)

Use this command to copy a Tivoli Storage Manager script to a new script.

### Privilege class

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

### Syntax

►► `COPY SCRIPT—current_script_name—new_script_name` ◀◀

### Parameters

*current\_script\_name* **(Required)**

Specifies the name of the script you want to copy.

*new\_script\_name* **(Required)**

Specifies the name of the new script. You can specify up to 30 characters for the name.

### Example: Make a copy of a script

Copy script TESTDEV to a new script and name it ENGDEV.

```
copy script testdev engdev
```

### Related commands

Table 56. Commands related to COPY SCRIPT

| Command       | Description  |
|---------------|--|
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| DELETE SCRIPT | Deletes the script or individual lines from the script.    |
| QUERY SCRIPT  | Displays information about scripts.                        |
| RENAME SCRIPT | Renames a script to a new name.                            |
| RUN           | Runs a script.   |
| UPDATE SCRIPT | Changes or adds lines to a script.                         |

## COPY SERVERGROUP (Copy a server group)

Use this command to create a copy of a server group.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► `COPY SERVERGroup—current_group_name—new_group_name` ◄◄

### Parameters

*current\_group\_name* **(Required)**

Specifies the server group to copy.

*new\_group\_name* **(Required)**

Specifies the name of the new server group. The maximum length of this name is 64 characters.

### Example: Make a copy of a server group

Copy the server group GRP\_PAYROLL to the new group HQ\_PAYROLL.

```
copy servergroup grp_payroll hq_payroll
```

### Related commands

Table 57. Commands related to COPY SERVERGROUP

| Command            | Description   |
|--------------------|---|
| DEFINE GRPMEMBER   | Defines a server as a member of a server group.       |
| DEFINE SERVER      | Defines a server for server-to-server communications. |
| DEFINE SERVERGROUP | Defines a new server group.                           |
| DELETE GRPMEMBER   | Deletes a server from a server group.                 |
| DELETE SERVER      | Deletes the definition of a server.                   |
| DELETE SERVERGROUP | Deletes a server group.                               |
| MOVE GRPMEMBER     | Moves a server group member.                          |
| QUERY SERVER       | Displays information about servers.                   |
| QUERY SERVERGROUP  | Displays information about server groups.             |
| RENAME SERVERGROUP | Renames a server group.                               |
| UPDATE SERVER      | Updates information about a server.                   |
| UPDATE SERVERGROUP | Updates a server group.                               |

## DEFINE commands

Use the DEFINE commands to create Tivoli Storage Manager objects.

The following is a list of DEFINE commands for Tivoli Storage Manager:

- “DEFINE ASSOCIATION (Associate client nodes with a schedule)” on page 106
- “DEFINE BACKUPSET (Define a backup set)” on page 108
- “DEFINE CLIENTACTION (Define a one-time client action)” on page 113
- “DEFINE CLIENTOPT (Define an option to an option set)” on page 118
- “DEFINE CLOPTSET (Define a client option set name)” on page 121
- “DEFINE COLLOGROUP (Define a collocation group)” on page 122
- “DEFINE COLLOCMEMBER (Define collocation group member)” on page 124
- “DEFINE COPYGROUP (Define a copy group)” on page 126
- “DEFINE DATAMOVER (Define a data mover)” on page 136
- “DEFINE DEVCLASS (Define a device class)” on page 139
- “DEFINE DOMAIN (Define a new policy domain)” on page 214
- “DEFINE DRIVE (Define a drive to a library)” on page 216
- “DEFINE EVENTSERVER (Define a server as the event server)” on page 220
- “DEFINE GRPMEMBER (Add a server to a server group)” on page 221
- “DEFINE LIBRARY (Define a library)” on page 222
- “DEFINE MACHINE (Define machine information for disaster recovery)” on page 231
- “DEFINE MACHNODEASSOCIATION (Associate a node with a machine)” on page 233
- “DEFINE MGMTCLASS (Define a management class)” on page 235
- “DEFINE NODEGROUP (Define a node group)” on page 238
- “DEFINE NODEGROUPMEMBER (Define node group member)” on page 239
- “DEFINE PATH (Define a path)” on page 240
- “DEFINE POLICYSET (Define a policy set)” on page 246
- “DEFINE PROFASSOCIATION (Define a profile association)” on page 248
- “DEFINE PROFILE (Define a profile)” on page 254
- “DEFINE RECMEDMACHASSOCIATION (Associate recovery media with a machine)” on page 256
- “DEFINE RECOVERYMEDIA (Define recovery media)” on page 258
- “DEFINE SCHEDULE (Define a client or an administrative command schedule)” on page 260
- “DEFINE SCRIPT (Define a Tivoli Storage Manager script)” on page 281
- “DEFINE SERVER (Define a server for server-to-server communications)” on page 283
- “DEFINE SERVERGROUP (Define a server group)” on page 288
- “DEFINE SPACETRIGGER (Define the space trigger)” on page 289
- “DEFINE STGPOOL (Define a storage pool)” on page 292
- “DEFINE SUBSCRIPTION (Define a profile subscription)” on page 333
- “DEFINE VIRTUALFSMAPPING (Define a virtual file space mapping)” on page 335
- “DEFINE VOLUME (Define a volume in a storage pool)” on page 337

### DEFINE ASSOCIATION (Associate client nodes with a schedule)

Use this command to associate one or more clients with a schedule. You must assign a client node to the policy domain to which a schedule belongs. Client nodes process operations according to the schedules associated with the nodes.

#### Note:

1. Tivoli Storage Manager cannot run multiple schedules concurrently for the same client node.
2. In a macro, the server may stall if some commands (such as 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.

#### Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege
- Restricted policy privilege for the policy domain to which the schedule belongs

#### Syntax

```

▶▶—DEfIne ASSOCiation—domain_name—schedule_name—node_name—▶▶

```

#### Parameters

##### *domain\_name* (Required)

Specifies the name of the policy domain to which the schedule belongs.

##### *schedule\_name* (Required)

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

##### *node\_name* (Required)

Specifies the name of a client node or a list of client nodes to associate with the specified schedule. Use commas to separate the items in the list. Do not leave spaces between the items and commas. You can use a wildcard character to specify a name. The command will not associate a listed client to the schedule if:

- The client is already associated with the specified schedule.
- The client is not assigned to the policy domain to which the schedule belongs.
- The client is a NAS node name. All NAS nodes are ignored.

#### Example: Associate client nodes with a schedule

Associate the client nodes SMITH or JOHN with the WEEKLY\_BACKUP schedule. The associated clients are assigned to the EMPLOYEE\_RECORDS policy domain.

```

define association employee_records
weekly_backup smith*,john*

```



## Example: Associate client nodes with a schedule

Associate the client nodes JOE, TOM, and LARRY with the WINTER schedule. The associated clients are assigned to the EMPLOYEE\_RECORDS policy domain; however, the client JOE is already associated with the WINTER schedule.

```
define association employee_records
winter joe,tom,larry
```

## Related commands

Table 58. Commands related to DEFINE ASSOCIATION

| Command            | Description   |
|--------------------|---|
| DEFINE SCHEDULE    | Defines a schedule for a client operation or an administrative command. |
| DELETE ASSOCIATION | Deletes the association between clients and a schedule.                 |
| DELETE SCHEDULE    | Deletes a schedule from the database.                                   |
| QUERY ASSOCIATION  | Displays the clients associated with one or more schedules.             |
| REGISTER NODE      | Defines a client to the server and sets options for that user.          |

### DEFINE BACKUPSET (Define a backup set)

Use this command to define a client backup set that was previously generated on one server and make it available to the server running this command. The client node has the option of restoring the backup set from the server running this command rather than the one on which the backup set was generated.

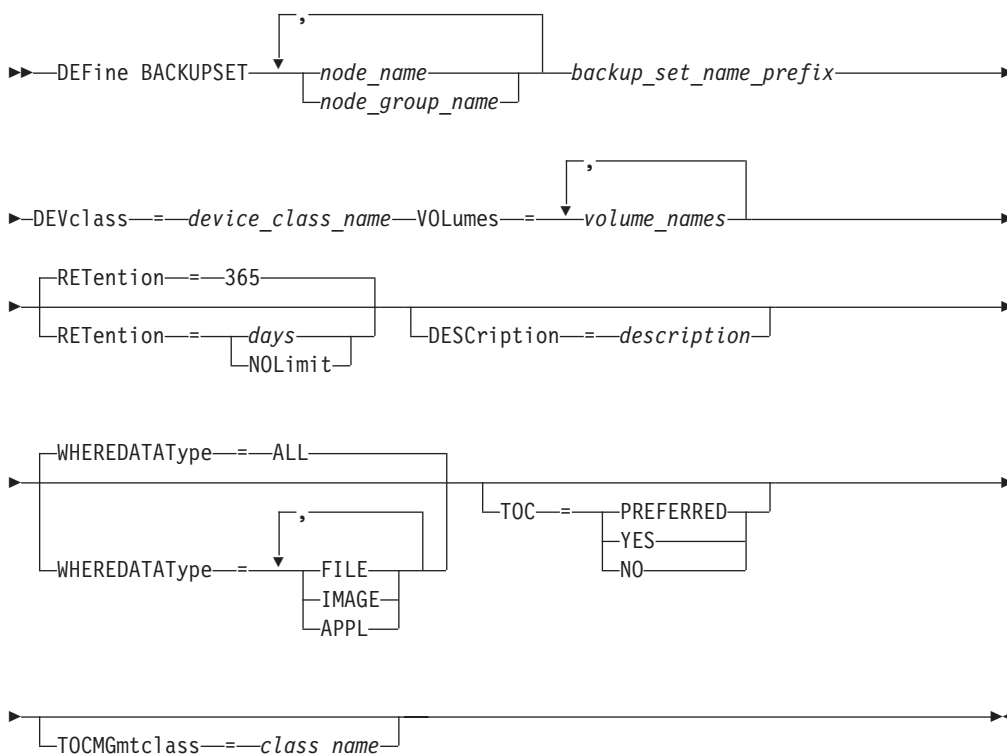
Any backup set generated on one server can be defined to another server as long as the servers share a common device type. The level of the server to which the backup set is being must be equal to or greater than the level of the server that generated the backup set.

You can also use the DEFINE BACKUPSET command to redefine a backup set that was deleted on a server.

### Privilege class

If the REQSYSAUTHOUTFILE server option is set to YES (the default), the administrator must have system privilege. If the REQSYSAUTHOUTFILE server option is set to NO, the administrator must have system privilege or policy privilege for the domain to which the client node is assigned.

### Syntax



### Parameters

#### *node\_name* or *node\_group\_name* (Required)

Specifies the name of the client nodes or node groups whose data is contained in the specified backup set volumes. To specify multiple node and node group

names, separate the names with commas and no intervening spaces. Node names can contain wildcard characters, but node group names cannot. If the backup set volumes contain backup sets from multiple nodes, every backup set whose node name matches one of the specified node names will be defined. If the volumes contain a backup set for a node that is not currently registered, the DEFINE BACKUPSET command will not define the backup set for that node.

### *backup\_set\_name\_prefix* (Required)

Specifies the name of the backup set to define to this server. The maximum length of the name is 30 characters.

When you select a name, Tivoli Storage Manager adds a suffix to construct the backup set name. For example, if you name your backup set *mybackupset*, Tivoli Storage Manager adds a unique number such as 3099 to the name. Your backup set name is then identified as *mybackupset.3099*. To later display information about this backup set, you can include a wildcard with the name, such as *mybackupset\** or you can specify the fully qualified name, such as *mybackupset.3099*.

If the backup set volumes contain backup sets for multiple nodes, then backup sets will be defined for each of the nodes using the same backup set name prefix and suffix.

### DEVclass (Required)

Specifies the device class name for the volumes from which the backup set is read.

**Note:** The device type associated with the device class you specify must match the device class with which the backup set was originally generated.

### VOLumes (Required)

Specifies the names of the volumes used to store the backup set. You can specify multiple volumes by separating the names with commas and no intervening spaces. The volumes you specify must be available to the server that is defining the backup set.

**Note:** The volumes you specify must be listed in the order they were created, or the DEFINE BACKUPSET command will fail.

The server does not verify that every volume specified for a multiple-volume backup set actually contains part of the backup set. The first volume is always checked, and in some cases additional volumes are also checked. If these volumes are correct, the backup set is defined and all of the volumes listed in the command are protected from being overwritten. If a volume that contains part of the backup set is not listed in the command, the volume will not be protected and can potentially be overwritten during normal server operations.

**Note:** By default, the server attempts to create a table of contents when a backup set is defined. If an incorrect volume is specified, or if volumes are not listed in the correct order, the table of contents creation will fail. If this occurs, check the volume list in the command and consider using the QUERY BACKUPSETCONTENTS command to verify the contents of the backup set.

### RETention

Specifies the number of days that the backup set is retained on the server. You can specify an integer from 0 to 30000. The default is 365 days. The values are:

*days*

Specifies the number of days to retain the backup set on the server.

## DEFINE BACKUPSET

### **NOLimit**

Specifies that the backup set should be retained on the server indefinitely.

If you specify **NOLIMIT**, Tivoli Storage Manager retains the volumes containing the backup set forever, unless a user or administrator deletes the volumes from server storage.

### **DEScRiption**

Specifies the description to associate with the backup set that belongs to the client node. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

### **WHEREDataTpe**

Specifies the backup sets containing the specified types of data are to be defined. This parameter is optional. The default is that backup sets for all types of data (file level, image, and application) are to be defined. To specify multiple data types, separate the data types with commas and no intervening spaces. Possible values are:

#### **ALL**

Specifies that backup sets for all types of data (file level, image, and application) are to be defined. This is the default.

#### **FILE**

Specifies that a file level backup set is to be defined. File level backup sets contain files and directories backup up by the backup-archive client.

#### **IMAGE**

Specifies that an image backup set is to be defined. Image backup sets contain images created by the backup-archive client **BACKUP IMAGE** command.

#### **APPL**

Specifies that an application backup set is to be defined. Application backup sets contain data backed up to the Tivoli Storage Manager server by applications using the Tivoli Storage Manager application programming interface. This parameter applies only if the Tivoli Storage Manager server was upgraded from Tivoli Storage Manager Express and only for the Tivoli Storage Manager Express versions of Data Protection for Exchange and Data Protection for SQL.

### **TOC**

Specifies whether a table of contents (TOC) should be created for the file level backup set when it is defined. The TOC parameter is ignored when defining image and application data backup sets because a table of contents is always created for these backup sets.

Consider the following in determining whether you want to create a table of contents:

- If a table of contents is created, you can use the Tivoli Storage Manager Web backup-archive client to examine the entire file system tree and choose files and directories to restore. Creation of a table of contents requires that you define the **TOCDESTINATION** attribute in the backup copy group for the management class specified by the **TOCMGMTCLASS** parameter. Note that a table of contents creation requires additional processing, storage pool space, and possibly a mount point during the backup set operation.

- If a table of contents is not saved for a backup set, you can still restore individual files or directory trees using the backup-archive client RESTORE BACKUPSET command, provided that you know the fully qualified name of each file or directory to be restored.

This parameter is optional. The default value is Preferred. Possible values are:

## No

Specifies that table of contents information is not saved for file level backup sets.

## Preferred

Specifies that table of contents information should be saved for file level backup sets. However, a backup set does not fail just because an error occurs during creation of the table of contents.

## Yes

Specifies that table of contents information must be saved for each file level backup set. A backup set fails if an error occurs during creation of the table of contents.

## TOCMgmtclass

Specifies the name of the management class to which the table of contents should be bound. If you do not specify a management class, the table of contents is bound to the default management class for the policy domain to which the node is assigned. In this case, creation of a table of contents requires that you define the TOCDESTINATION attribute in the backup copy group for the specified management class.

## Example: Define a backup set

Define the PERS\_DATA backup set that belongs to client node JANE to the server running this command. Retain the backup set on the server for 50 days. Specify that volumes VOL001 and VOL002 contain the data for the backup set. The volumes are to be read by a device that is assigned to the AGADM device class. Include a description.

```
define backupset jane pers_data devclass=agadm
volumes=vol1,vol2 retention=50
description="sector 7 base image"
```

## Related commands

Table 59. Commands related to DEFINE BACKUPSET

| Command                 | Description                                     |
|-------------------------|---|
| DEFINE NODEGROUP        | Defines a group of nodes.                       |
| DEFINE NODEGROUPMEMBER  | Adds a client node to a node group.             |
| DELETE NODEGROUP        | Deletes a node group.                           |
| DELETE BACKUPSET        | Deletes a backup set.                           |
| DELETE NODEGROUPMEMBER  | Deletes a client node from a node group.        |
| GENERATE BACKUPSET      | Generates a backup set of a client's data.      |
| GENERATE BACKUPSETTOC   | Generates a table of contents for a backup set. |
| QUERY BACKUPSET         | Displays backup sets.                           |
| QUERY BACKUPSETCONTENTS | Displays contents contained in backup sets.     |
| QUERY NODEGROUP         | Displays information about node groups.         |

## DEFINE BACKUPSET

*Table 59. Commands related to DEFINE BACKUPSET (continued)*

| Command          | Description   |
|------------------|---|
| UPDATE BACKUPSET | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP | Updates the description of a node group.                |

## DEFINE CLIENTACTION (Define a one-time client action)

Use this command to schedule one or more clients to process a command for a one-time action.

The server automatically defines a schedule and associates the client node to the schedule. The server assigns the schedule priority 1, sets the PERUNITS to ONETIME, and determines the number of days to keep the schedule active. The number of days is based on the value set with the SET CLIENTACTDURATION command.

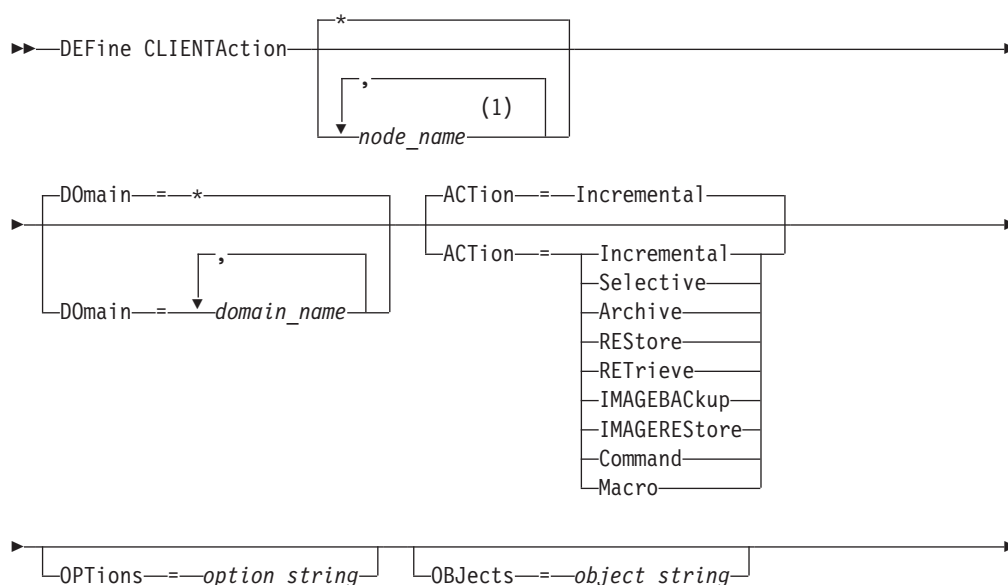
How quickly the client processes this command depends on whether the scheduling mode for the client is set to server-prompted or client-polling. The client scheduler must be started on the client workstation in order for the server to process the schedule.

**Remember:** The start of the Tivoli Storage Manager scheduler depends on the processing of other threads in the server and other processes on the Tivoli Storage Manager server host system. The amount of time it takes to start the scheduler also depends on network traffic and how long it takes to open a socket, to connect with the Tivoli Storage Manager client, and to receive a response from the client. In general, the greater the processing and connectivity requirements on the Tivoli Storage Manager server and client, the longer it can take to start the scheduler.

### Privilege class

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

### Syntax



## DEFINE CLIENTACTION



### Notes:

- 1 If you explicitly specify a NAS node name, the command will fail. If you provide a pattern-matching expression for the node, any NAS nodes that match the pattern will be ignored.

### Parameters

#### *node\_name*

Specifies the list of client nodes that will process the schedule associated with the action. You can use a wildcard character to specify a client node or a list of client nodes. Separate the client node names with commas and no intervening spaces. If you do not specify a value, all client nodes will be scheduled for the action.

#### **Domain**

Specifies the list of policy domains used to limit the list of client nodes. Only client nodes that are assigned to one of the specified policy domains will be scheduled. All clients assigned to a matching domain will be scheduled. Separate multiple domain names with commas and no intervening spaces. If you do not specify a value, all policy domains will be included in the list.

#### **ACTion**

Specifies the action that occurs when this schedule is processed. Possible values are:

##### **Incremental**

Specifies that the schedule backs up all files that are new or that have changed since the last incremental backup. Incremental also backs up any file for which all existing backups may have expired.

##### **Selective**

Specifies that the schedule backs up only files that are specified with the OBJECTS parameter.

##### **Archive**

Specifies that the schedule archives files that are specified with the OBJECTS parameter.

##### **REStore**

Specifies that the schedule restores files that are specified with the OBJECTS parameter.

When you specify ACTION=RESTORE for a scheduled operation, and the REPLACE option is set to PROMPT, no prompting occurs. If you set the option to PROMPT, the files are skipped.

If you specify a second file specification, this second file specification acts as the restore destination. If you need to restore multiple groups of files, schedule one for each file specification that you need to restore.

##### **REtrieve**

Specifies that the schedule retrieves files that are specified with the OBJECTS parameter.



### IMAGEBACKup

Specifies that the schedule backs up logical volumes that are specified with the OBJECTS parameter.

### IMAGERESTore

Specifies that the schedule restores logical volumes that are specified with the OBJECTS parameter.

### Command

Specifies that the schedule processes a client operating system command or script that is specified with the OBJECTS parameter.

### Macro

Specifies that a client processes a macro whose file name is specified with the OBJECTS parameter.

### OPTions

Specifies the client options that you specify to the scheduled command at the time the schedule is processed. This parameter is optional.

Only those options that are valid on the scheduled command may be specified for this parameter. Refer to the appropriate client manual for information about options that are valid from the command line. All options described there as valid only on the initial command line will either result in an error or simply be ignored when the schedule from the server is run. For example, do not include the following options because they have no effect when the client processes the scheduled command:

MAXCMDRETRIES  
OPTFILE  
QUERYSCHEDPERIOD  
RETRYPERIOD  
SCHEDLOGNAME  
SCHEDMODE  
SERVERNAME  
TCPCLIENTADDRESS  
TCPCLIENTPORT

If the option string contains multiple options or options with embedded spaces, surround the entire option string with one pair of single quotes. Individual options that contain spaces should be enclosed in double quotes. A leading minus sign is required in front of the option. Errors can occur if the option string contains spaces that are not quoted correctly.

The following examples show how to specify some client options:

- To specify subdir=yes and domain all-local -systemobject, enter:  
options='-subdir=yes -domain="all-local -c: -systemobject"'
- To specify domain all-local -c: -d:, enter:  
options='-domain="all-local -c: -d:"'

### OBjects

Specifies the objects for which the specified action is performed. Use a single space between each object. This parameter is required except when ACTION=INCREMENTAL. If the action is a backup, archive, retrieve, or restore operation, the objects are file spaces, directories, or logical volumes. See the *Backup-Archive Clients Installation and User's Guide* for command syntax information. If the action is to run a command or macro, the object is the name of the command or macro to run.

## DEFINE CLIENTACTION

When you specify ACTION=INCREMENTAL without specifying a value for this parameter, the scheduled command is invoked without specified objects and attempts to process the objects as defined in the client's option file. To select all file spaces or directories for an action, explicitly list them in the object string. Entering only an asterisk in the object string causes the backup to occur only for the directory where the scheduler was started.

### Important:

- If you specify a second file specification, and it is not a valid destination, you receive this error:  
ANS1082E Invalid destination file specification <filespec> entered.
- If you specify more than two file specifications, you receive this error:  
ANS1102E Excessive number of command line arguments passed to the program!

When you specify ACTION=ARCHIVE, INCREMENTAL, or SELECTIVE for this parameter, you can list a maximum of twenty (20) file specifications.

Enclose the object string in double quotes if it contains blank characters (spaces), and then surround the double quotes with single quotes. If the object string contains multiple file names, enclose each file name with its own pair of double quotes, then surround the entire string with one pair of single quotes. Errors can occur if file names contain a space that is not quoted correctly. The following examples show how to specify some file names:

- To specify /home/file 2, /home/gif files, and /home/my test file, enter:  
OBJECTS="'/home/file 2' '/home/gif files' '/home/my test file'"
- To specify /home/test file, enter:  
OBJECTS="'/home/test file'"

### Wait

Specifies whether to wait for a scheduled client operation to complete. This parameter is useful when defining client actions from a command script or macro. This parameter is optional. The default is No. Possible values are:

#### No

Specifies that you do not wait for the scheduled client operation to complete. If you specify this value and the value of the ACTION parameter is COMMAND, the return code indicates whether the client action was defined.

#### Yes

Specifies that you wait for the scheduled client operation to complete. If you specify this value and the value of the ACTION parameter is COMMAND, the return code indicates the status of the client operation.

You cannot issue the DEFINE CLIENTACTION command with WAIT=YES from the server console. However, from the server console, you can:

- Specify WAIT=YES with DEFINE CLIENTACTION as the command line of a DEFINE SCRIPT command.
- Specify WAIT=YES with DEFINE CLIENTACTION as the command line of a file whose contents will be read into the script that is defined by a DEFINE SCRIPT command.

**Restriction:** If you specify the DEFINE CLIENTACTION command with WAIT=YES in a macro, the immediate schedules defined by the command will not roll back if the macro does not complete successfully.

## Example: Perform a one-time incremental backup

Issue an incremental backup command for client node TOM assigned to policy domain EMPLOYEE\_RECORDS. Tivoli Storage Manager defines a schedule and associates the schedule to client node TOM (assuming that the client scheduler is running).

```
define clientaction tom domain=employee_records
action=incremental
```

## Related commands

Table 60. Commands related to DEFINE CLIENTACTION

| Command               | Description   |
|-----------------------|---|
| DELETE SCHEDULE       | Deletes a schedule from the database.   |
| QUERY ASSOCIATION     | Displays the clients associated with one or more schedules.                         |
| QUERY EVENT           | Displays information about scheduled and completed events for selected clients.     |
| QUERY SCHEDULE        | Displays information about schedules.   |
| SET CLIENTACTDURATION | Specifies the duration of a schedule defined using the DEFINE CLIENTACTION command. |

## DEFINE CLIENTOPT (Define an option to an option set)

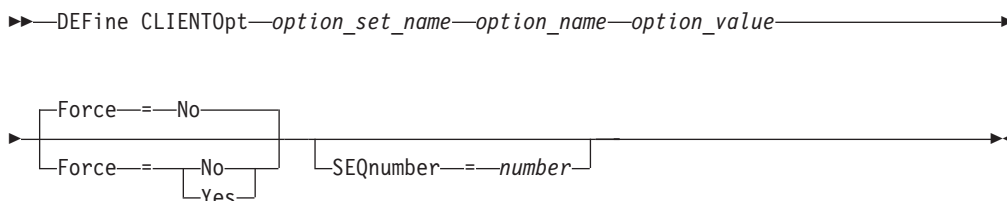
Use this command to add a client option to an option set.

For details about the options and the values you can specify, refer to *Backup-Archive Clients Installation and User's Guide*.

### Privilege class

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

### Syntax



### Parameters

#### *option\_set\_name* (Required)

Specifies the name of the option set.

#### *option\_name* (Required)

Specifies a client option to add to the option set.

For a list of valid client options, see "Client options that can be set by the Tivoli Storage Manager server" at the Tivoli Storage Manager information center ([http://publib.boulder.ibm.com/infocenter/tsminfo/v6/topic/com.ibm.itsm.client.doc/c\\_opt\\_setbyserver.html](http://publib.boulder.ibm.com/infocenter/tsminfo/v6/topic/com.ibm.itsm.client.doc/c_opt_setbyserver.html)).

**Note:** To define include-exclude values, specify the include or exclude option with *option-name*, and use *option\_value* to specify any valid include or exclude statement, as you would in the client options file. For example:

```
define clientopt option_set_name inclexcl "include c:\proj\text\devel.*"
```

#### *option\_value* (Required)

Specifies the value for the option. If the option includes more than one value, enclose the value in quotation marks. For the values you can specify with the option refer to *Backup-Archive Clients Installation and User's Guide*.

#### Note:

1. The QUIET and VERBOSE options do not have an option value in the client option's file. To specify these values in a server client option set, specify a value of YES or NO.
2. To add an INCLUDE or EXCLUDE option for a file name that contains one or more spaces, put single quotation marks around the file specification, and double quotation marks around the entire option. See "Example: Add an option to a client option set" on page 119 for more information.
3. The *option\_value* is limited to 1024 characters.

#### Force

Specifies whether the server forces the client to use the option set value. The

value is ignored for additive options, such as INCLEXCL and DOMAIN. The default is NO. This parameter is optional. The values are:

**Yes**

Specifies that the server forces the client to use the value. (The client cannot override the value.)

**No**

Specifies that the server does not force the client to use the value. (The client can override the value.)

**SEQnumber**

Specifies a sequence number when an option name is specified more than once. This parameter is optional.

**Example: Add an option to a client option set**

Add a client option (MAXCMDRETRIES 5) to a client option set named ENG.  

```
define clientopt eng maxcmdretries 5
```

**Example: Add an option to exclude a file from backup**

Add a client option to the option set ENGBACKUP to exclude the c:\admin\file.txt from backup services.  

```
define clientopt engbackup inclexcl "exclude c:\admin\file.txt"
```

**Example: Add an option to exclude a directory from backup**

Add a client option to the option set WINSPEC to exclude a temporary internet directory from backup services. When you use the EXCLUDE or INCLUDE option with file names that contain spaces, put single quotation marks around the file specification, then double quotation marks around the entire option.  

```
define clientopt winspec inclexcl "exclude.dir '*:\...\Temporary Internet Files'"
```

**Example: Add an option to bind files in specified directories**

Add client options to the option set WINSPEC to bind all files in directories C:\Data and C:\Program Files\My Apps to a management class named PRODCCLASS.  

```
define clientopt winspec inclexcl "include C:\Data\...\* prodclass"
define clientopt winspec inclexcl "include 'C:\Program Files\My Apps\...\*' prodclass"
```

**Related commands**

*Table 61. Commands related to DEFINE CLIENTOPT*

| Command          | Description  |
|------------------|--|
| COPY CLOPTSET    | Copies a client option set.                                    |
| DEFINE CLOPTSET  | Defines a client option set.                                   |
| DELETE CLIENTOPT | Deletes a client option from a client option set.              |
| DELETE CLOPTSET  | Deletes a client option set.                                   |
| REGISTER NODE    | Defines a client to the server and sets options for that user. |

## DEFINE CLIENTOPT

*Table 61. Commands related to DEFINE CLIENTOPT (continued)*

| Command          | Description  |
|------------------|--|
| QUERY CLOPTSET   | Displays information about a client option set.                        |
| UPDATE CLIENTOPT | Updates the sequence number of a client option in a client option set. |
| UPDATE CLOPTSET  | Updates the description of a client option set.                        |
| UPDATE NODE      | Changes the attributes associated with a client node.                  |

## DEFINE CLOPTSET (Define a client option set name)

Use this command to define a name for a set of options you can assign to clients for archive, backup, restore, and retrieve operations.

To add options to the new set, issue the DEFINE CLIENTOPT command.

### Privilege class

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

### Syntax

```
►►—DEFine CLOptset—option_set_name—┐
                                   └DESCription—=description—┘►►
```

### Parameters

#### *option\_set\_name* (Required)

Specifies the name of the client option set. The maximum length of the name is 64 characters.

#### DESCription

Specifies a description of the client option set. The maximum length of the description is 255 characters. The description must be enclosed in quotation marks if it contains any blank characters. This parameter is optional.

### Example: Define a client option set

To define a client option set named ENG issue the following command.

```
define cloptset eng
```

### Related commands

Table 62. Commands related to DEFINE CLOPTSET

| Command          | Description  |
|------------------|--|
| COPY CLOPTSET    | Copies a client option set.  |
| DEFINE CLIENTOPT | Adds a client option to a client option set.                           |
| DELETE CLIENTOPT | Deletes a client option from a client option set.                      |
| DELETE CLOPTSET  | Deletes a client option set.   |
| QUERY CLOPTSET   | Displays information about a client option set.                        |
| UPDATE CLIENTOPT | Updates the sequence number of a client option in a client option set. |
| UPDATE CLOPTSET  | Updates the description of a client option set.                        |

## DEFINE COLLOGROUP (Define a collocation group)

Use this command to define a collocation group. A *collocation group* is a group of nodes whose data is collocated on a minimal number of sequential access volumes. Their data is collocated only if the storage pool definition is set to collocate by group (COLLOCATE=GROUP).

### Privilege class

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

### Syntax

```
►►—DEfINE COLLOCGroup—group_name—[DESCRiption==description]—►►
```

### Parameters

*group\_name*

Specifies the name of the collocation group name that you want to create. The maximum length of the name is 30 characters.

DESCRiption

Specifies a description of the collocation group. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

### Example: Define a collocation group

To define a collocation group named GROUP1 issue the following command:

```
define collogroup group1
```

### Related commands

Table 63. Commands related to DEFINE COLLOGROUP

| Command             | Description  |
|---------------------|--|
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                       |
| DEFINE STGPOOL      | Defines a storage pool as a named collection of server storage media.            |
| DELETE COLLOGROUP   | Deletes a collocation group.   |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                  |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces.    |
| QUERY COLLOGROUP    | Displays information about collocation groups.                                   |
| QUERY NODE          | Displays partial or complete information about one or more clients.              |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.      |
| QUERY STGPOOL       | Displays information about storage pools.  |
| REMOVE NODE         | Removes a client from the list of registered nodes for a specific policy domain. |



*Table 63. Commands related to DEFINE COLLOCGROUP (continued)*

| <b>Command</b>     | <b>Description</b>                              |
|--------------------|---|
| UPDATE COLLOCGROUP | Updates the description of a collocation group. |
| UPDATE STGPOOL     | Changes the attributes of a storage pool.       |

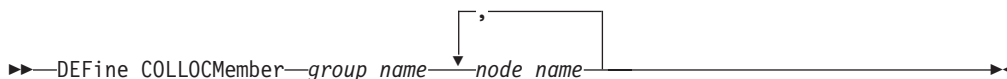
## DEFINE COLLOCMEMBER (Define collocation group member)

Use this command to add a client node to a collocation group. A *collocation group* is a group of nodes whose data is collocated on a minimal number of sequential access volumes.

### Privilege class

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

### Syntax



### Parameters

*group\_name*

Specifies the name of the collocation group to which you want to add a client node.

*node\_name*

Specifies the name of the client node that you want to add to the collocation group. You can specify one or more names. Separate multiple names with commas; do not use intervening spaces. You can also use wildcard characters when specifying multiple names.

### Example: Define two collocation group members

Define two members, NODE1 and NODE2, to a collocation group, GROUP1.

```
define collocmember group1 node1,node2
```

### Related commands

Table 64. Commands related to DEFINE COLLOCMEMBER

| Command             | Description   |
|---------------------|---|
| DEFINE COLLOGROUP   | Defines a collocation group.  |
| DEFINE STGPPOOL     | Defines a storage pool as a named collection of server storage media.         |
| DELETE COLLOGROUP   | Deletes a collocation group.  |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                               |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces. |
| QUERY COLLOGROUP    | Displays information about collocation groups.                                |
| QUERY NODE          | Displays partial or complete information about one or more clients.           |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.   |
| QUERY STGPPOOL      | Displays information about storage pools.                                     |

*Table 64. Commands related to DEFINE COLLOCMEMBER (continued)*

| Command            | Description  |
|--------------------|--|
| REMOVE NODE        | Removes a client from the list of registered nodes for a specific policy domain. |
| UPDATE COLLOCGROUP | Updates the description of a collocation group.                                  |
| UPDATE STGPOOL     | Changes the attributes of a storage pool.  |

### DEFINE COPYGROUP (Define a copy group)

Use this command to define a new backup or archive copy group within a specific management class, policy set, and policy domain. The server uses the backup and archive copy groups to control how clients back up and archive files, and to manage the backed-up and archived files.

To allow clients to use the new copy group, you must activate the policy set that contains the new copy group.

You can define one backup and one archive copy group for each management class. To ensure that client nodes can back up files, include a backup copy group in the default management class for a policy set.

**Attention:** The DEFINE COPYGROUP command fails if you specify a copy storage pool as a destination.

The DEFINE COPYGROUP command has two forms, one for defining a backup copy group and one for defining an archive copy group. The syntax and parameters for each form are defined separately.

*Table 65. Commands related to DEFINE COPYGROUP*

| Command                        | Description   |
|--------------------------------|---|
| ASSIGN DEFMGMTCLASS            | Assigns a management class as the default for a specified policy set.               |
| BACKUP NODE                    | Backs up a network-attached storage (NAS) node.                                     |
| COPY MGMTCLASS                 | Creates a copy of a management class.   |
| DEFINE MGMTCLASS               | Defines a management class.   |
| DEFINE STGPOOL                 | Defines a storage pool as a named collection of server storage media.               |
| DELETE COPYGROUP               | Deletes a backup or archive copy group from a policy domain and policy set.         |
| DELETE MGMTCLASS               | Deletes a management class and its copy groups from a policy domain and policy set. |
| EXPIRE INVENTORY               | Manually starts inventory expiration processing.                                    |
| QUERY COPYGROUP                | Displays the attributes of a copy group.  |
| QUERY MGMTCLASS                | Displays information about management classes.                                      |
| SET ARCHIVERETENTIONPROTECTION | Specifies whether data retention protection is activated.                           |
| UPDATE COPYGROUP               | Changes one or more attributes of a copy group.                                     |

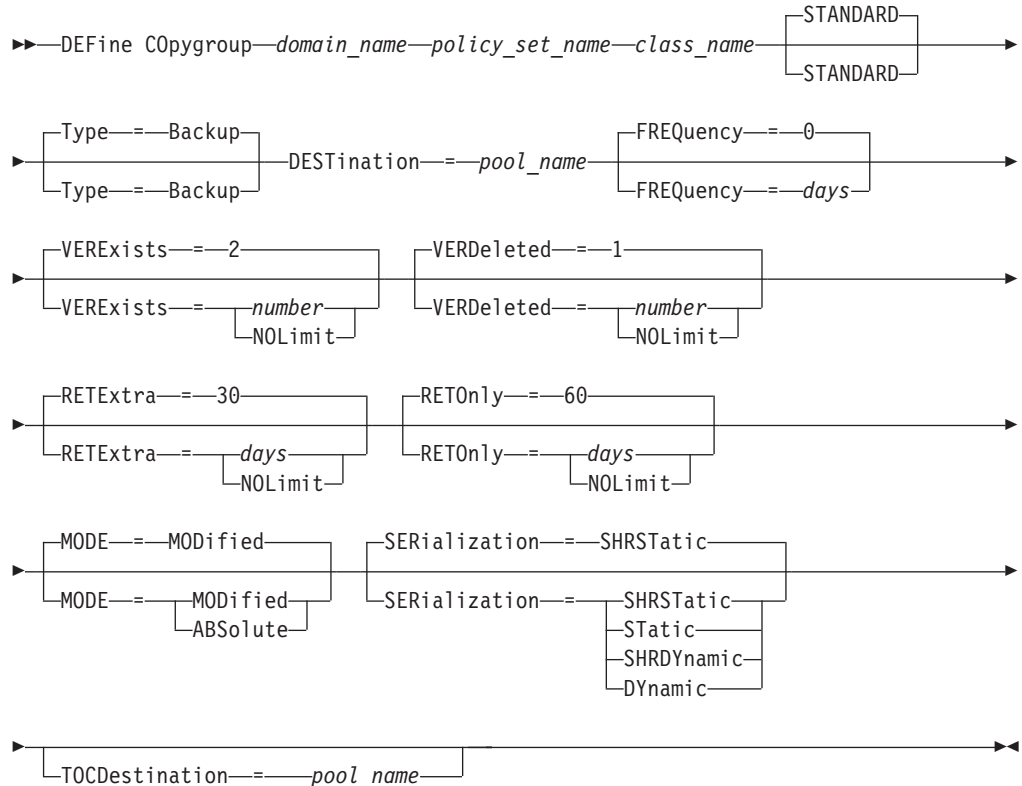
## DEFINE COPYGROUP (Define a backup copy group)

Use this command to define a new backup copy group within a specific management class, policy set, and policy domain.

### 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

#### *domain\_name* (Required)

Specifies the policy domain for which you are defining the copy group.

#### *policy\_set\_name* (Required)

Specifies the policy set for which you are defining the copy group.

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

#### *class\_name* (Required)

Specifies the management class for which you are defining the copy group.

#### STANDARD

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

## DEFINE COPYGROUP–backup

### **Type=Backup**

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

### **DESTination (Required)**

Specifies the primary storage pool where the server initially stores backup data. You cannot specify a copy storage pool as the destination.

### **FREQuency**

Specifies how frequently Tivoli Storage Manager can back up a file. This parameter is optional. Tivoli Storage Manager backs up a file only when the specified number of days has elapsed since the last backup. The FREQUENCY value is used only during a full incremental backup operation. This value is ignored during selective backup or partial incremental backup. You can specify an integer from 0 to 9999. The default value is 0, meaning that Tivoli Storage Manager can back up a file regardless of when the file was last backed up.

### **VERExists**

Specifies the maximum number of backup versions to retain for files that are currently on the client file system. This parameter is optional. The default value is 2.

If an incremental backup operation causes the limit to be exceeded, the server expires the oldest backup version that exists in server storage. Possible values are:

*number*

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 to 9999.

### **NOLimit**

Specifies that you want the server to retain all backup versions.

The number of backup versions to retain is controlled by this parameter until versions exceed the retention time specified by the RETEXTRA parameter.

### **VERDeleted**

Specifies the maximum number of backup versions to retain for files that have been deleted from the client file system after being backed up using Tivoli Storage Manager. This parameter is optional. The default value is 1.

If a user deletes a file from the client file system, the next incremental backup causes the server to expire the oldest versions of the file in excess of this number. The expiration date for the remaining versions is determined by the retention time specified by the RETEXTRA or RETONLY parameter. Possible values are:

*number*

Specifies the number of backup versions to retain for files that are deleted from the client file system after being backed up. You can specify an integer from 0 to 9999.

### **NOLimit**

Specifies that you want the server to retain all backup versions for files that are deleted from the client file system after being backed up.

### **RETEExtra**

Specifies the number of days to retain a backup version after that version becomes inactive. A version of a file becomes inactive when the client stores a more recent backup version, or when the client deletes the file from the workstation and then runs a full incremental backup. The server deletes

inactive versions based on retention time even if the number of inactive versions does not exceed the number allowed by the VEREXISTS or VERDELETED parameters. This parameter is optional. The default value is 30 days. Possible values are:

*days*

Specifies the number of days to retain inactive backup versions. You can specify an integer from 0 to 9999.

**NOLimit**

Specifies that you want to retain inactive backup versions indefinitely.

If you specify NOLIMIT, the server deletes inactive backup versions based on the VEREXISTS parameter (when the file still exists on the client file system) VERDELETED parameter (when the file no longer exists on the client file system).

**RETOOnly**

Specifies the number of days to retain the last backup version of a file that has been deleted from the client file system. This parameter is optional. The default value is 60. Possible values are:

*days*

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

**NOLimit**

Specifies that you want to keep the last remaining inactive version of a file indefinitely.

If you specify NOLIMIT, the server retains the last remaining backup version forever, unless a user or administrator deletes the file from server storage.

**MODE**

Specifies whether Tivoli Storage Manager backs up a file only if the file has changed since the last backup, or whenever a client requests a backup. This parameter is optional. The default value is MODIFIED. Possible values are:

**MODified**

Specifies that Tivoli Storage Manager backs up the file only if it has changed since the last backup. Tivoli Storage Manager considers a file 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 Tivoli Storage Manager backs up the file regardless of whether it has been modified.

The MODE value is used only for full incremental backup. This value is ignored during partial incremental backup or selective backup.

**SERialization**

Specifies how Tivoli Storage Manager processes files or directories when they are modified during backup processing. This parameter is optional. The default value is SHRSTATIC. Possible values are:

## DEFINE COPYGROUP–backup

### SHRStatic

Specifies that Tivoli Storage Manager backs up a file or directory only if it is not being modified during backup. Tivoli Storage Manager attempts to perform a backup as many as four times, depending on the value specified for the CHANGINGRETRIES client option. If the file or directory is modified during each backup attempt, Tivoli Storage Manager does not back it up.

### Static

Specifies that Tivoli Storage Manager backs up a file or directory only if it is not being modified during backup. Tivoli Storage Manager attempts to perform the backup only once.

Platforms that do not support the STATIC option default to SHRSTATIC.

### SHRDynamic

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

### Dynamic

Specifies that Tivoli Storage Manager backs up a file or directory on the first attempt, regardless of whether the file or directory is being modified during backup processing.

**Attention:** Be careful about using the SHRDYNAMIC and DYNAMIC values. Tivoli Storage Manager uses these values to determine if it backs up a file or directory while modifications are occurring. As a result, the backup version might be a fuzzy backup. A fuzzy backup does not accurately reflect what is currently in the file or directory because it contains some, but not all, modifications. If a file that contains a fuzzy backup is restored, the file may or may not be usable, depending on the application that uses the file. If a fuzzy backup is not acceptable, set `SERIALIZATION` to `SHRSTATIC` or `STATIC` so that Tivoli Storage Manager creates a backup version only if the file or directory is not being modified.

### TOCDestination

Specifies the primary storage pool in which a table of contents (TOC) will initially be stored for any Network Data Management Protocol (NDMP) backup or backup set operation for which a TOC is generated. This parameter is optional. You cannot specify a copy storage pool as the destination. The storage pool specified for the destination must have `NATIVE` or `NONBLOCK` data format. To avoid mount delays, it is recommended that the storage pool have a device class of `DISK` or `DEVTYPE=FILE`. TOC generation is an option for NDMP backup operations, but is not supported for other image-backup operations.

If TOC creation is requested for a backup operation that uses NDMP and the image is bound to a management class whose backup copy group does not specify a TOC destination, the outcome will depend on the TOC parameter for the backup operation.

- If `TOC=PREFERRED` (the default), the backup proceeds without creation of a TOC.
- If `TOC=YES`, the entire backup fails because no TOC can be created.



**Example: Create a backup copy group**

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

```
define copygroup employee_records  
vacation activefiles standard type=backup  
destination=backuppools frequency=3  
verexists=5 mode=absolute
```

## DEFINE COPYGROUP–archive

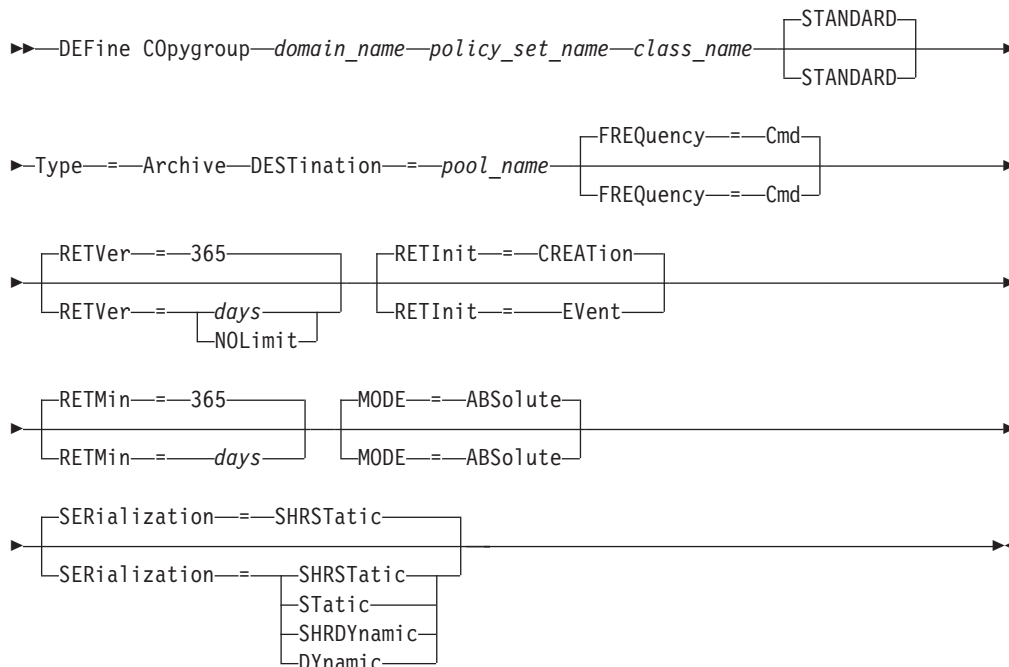
### DEFINE COPYGROUP (Define an archive copy group)

Use this command to define a new archive copy group within a specific management class, policy set, and policy domain.

#### 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

##### *domain\_name* (Required)

Specifies the name of the policy domain for which you are defining the copy group.

##### *policy\_set\_name* (Required)

Specifies the name of the policy set for which you are defining the copy group.

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

##### *class\_name* (Required)

Specifies the name of the management class for which you are defining the copy group.

##### STANDARD

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

##### Type=Archive (Required)

Specifies that you want to define an archive copy group.

**DESTination (Required)**

Specifies the primary storage pool where the server initially stores the archive copy. You cannot specify a copy storage pool as the destination.

**FREQuency=Cmd**

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

**RETVer**

Specifies the number of days to keep an archive copy. This parameter is optional. The default value is 365. Possible values are:

*days*

Specifies the number of days to keep an archive copy. You can specify an integer from 0 to 30000.

If you specify a value of 0 (zero), and the destination storage pool for the archive copy group is a Snaplock storage pool (RECLAMATIONTYPE=SNAPLOCK), retention of the volumes will be set using the value of the server option RETENTIONEXTENSION. The default value for the RETENTIONEXTENSION option is 365 (days).

**NOLimit**

Specifies that you want to keep an archive copy indefinitely.

If you specify NOLIMIT, the server retains archive copies forever, unless a user or administrator deletes the file from server storage. If you specify NOLIMIT, you cannot also specify EVENT for the **RETINIT** parameter.

The value of the **RETVER** parameter can affect the management class to which the server binds an archived directory. If the client does not use the ARCHMC option, the server binds directories that are archived to the default management class. If the default management class has no archive copy group, the server binds directories that are archived to the management class with the shortest retention period.

The **RETVER** parameter of the archive copy group of the management class to which an object is bound determines the retention criterion for each object. See the SET ARCHIVERETENTIONPROTECTION command for a description of data protection.

If the primary storage pool specified in the **DESTINATION** parameter belongs to a Centera device class and data protection has been enabled, then the RETVER value will be sent to Centera for retention management purposes. See the SET ARCHIVERETENTIONPROTECTION command for a description of data protection.

**RETInit**

Specifies when the retention time specified by the RETVER attribute is initiated. This parameter is optional. If you define the RETINIT value during copy group creation, you cannot modify it at a later time. The default value is CREATION. Possible values are:

**CREATion**

Specifies that the retention time specified by the RETVER attribute is initiated at the time an archive copy is stored on the Tivoli Storage Manager server.

**EVent**

Specifies that the retention time specified in the **RETVER** parameter is initiated at the time a client application notifies the server of a

## DEFINE COPYGROUP–archive

retention-initiating event for the archive copy. If you specify RETINIT=EVENT, you cannot also specify RETVER=NOLIMIT.

**Tip:** You can place a deletion hold on an object that was stored with RETINIT=EVENT for which the event has not been signaled. If the event is signaled while the deletion hold is in effect, the retention period will be initiated, but the object will not be deleted while the hold is in effect.

### RETRMin

Specifies the minimum number of days to keep an archive copy after it has been archived. This parameter is optional. The default value is 365. If you specify RETINIT=CREATION, this parameter is ignored.

### MODE=ABSolute

Specifies that a file is always archived when the client requests it. The MODE must be ABSOLUTE. This parameter is optional. The default value is ABSOLUTE.

### SERialization

Specifies how Tivoli Storage Manager processes files that are modified during archive. This parameter is optional. The default value is SHRSTATIC. Possible values are:

#### SHRStatic

Specifies that Tivoli Storage Manager archives a file only if it is not being modified. Tivoli Storage Manager attempts to perform an archive operation as many as four times, depending on the value specified for the CHANGINGRETRIES client option. If the file is modified during the archive attempt, Tivoli Storage Manager does not archive the file.

#### Static

Specifies that Tivoli Storage Manager archives a file only if it is not being modified. Tivoli Storage Manager attempts to perform the archive operation only once.

Platforms that do not support the STATIC option default to SHRSTATIC.

#### SHRDYnamic

Specifies that if the file is being modified during an archive attempt, Tivoli Storage Manager archives the file during its last attempt even though the file is being modified. Tivoli Storage Manager attempts to archive the file as many as four times, depending on the value specified for the CHANGINGRETRIES client option.

#### DYnamic

Specifies that Tivoli Storage Manager archives a file on the first attempt, regardless of whether the file is being modified during archive processing.

**Attention:** Be careful about using the SHRDYNAMIC and DYNAMIC values. Tivoli Storage Manager uses them to determine if it archives a file while modifications are occurring. As a result, the archive copy might be a fuzzy backup. A fuzzy backup does not accurately reflect what is currently in the file because it contains some, but not all, modifications. If a file that contains a fuzzy backup is retrieved, the file may or may not be usable, depending on the application that uses the file. If a fuzzy backup is not acceptable, set SERIALIZATION to SHRSTATIC or STATIC so that Tivoli Storage Manager creates an archive copy only if the file is not being modified.

**Example: Define an archive copy group for event-based retention**

Create an archive copy group named STANDARD for management class EVENTMC in policy set SUMMER in the PROG1 policy domain. Set the archive destination to ARCHIVEPOOL, where the archive copy is kept until the server is notified of an event to initiate the retention time, after which the archive copy is kept for 30 days. The archive copy will be kept for a minimum of 90 days after being stored on the server, regardless of when the server is notified of an event to initiate the retention time.

```
define copygroup prog1 summer eventmc standard type=archive
destination=archivepool retinit=event retver=30 retmin=90
```

## DEFINE DATAMOVER (Define a data mover)

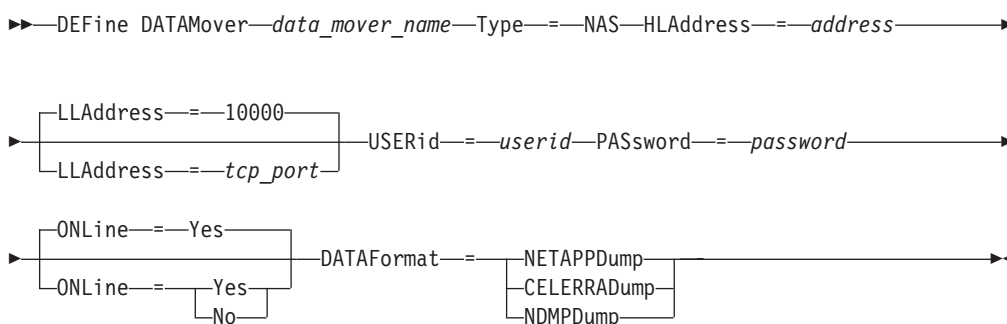
Use this command to define a data mover. A data mover is a named device that accepts a request from Tivoli Storage Manager to transfer data and can be used to perform outboard copy operations.

See the documentation for your device for guidance on specifying the parameters for this command.

### Privilege class

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

### Syntax



### Parameters

#### *data\_mover\_name* (Required)

Specifies the name of the data mover. This name must be the same as a node name that you previously registered using the `REGISTER NODE TYPE=NAS` command. The data that is backed up from this NAS data mover will be assigned to this node name in the server database. A maximum of 64 characters can be used to specify the name.

#### **Type=NAS** (Required)

Specifies that the data mover is a NAS file server.

#### **HLAddress** (Required)

Specifies either the numerical IP address or the domain name, which are used to access the NAS file server.

#### **LLAddress**

Specifies the TCP port number to access the NAS device for Network Data Management Protocol (NDMP) sessions. This parameter is optional. The default value is 10000.

#### **USERid** (Required)

Specifies the user ID for a user that is authorized to initiate an NDMP session with the NAS file server. For example, enter the administrative ID for a NetApp file server.

#### **PASsword** (Required)

Specifies the password for the user ID to log onto the NAS file server.

## ONLine

Specifies whether the data mover is available for use. This parameter is optional. The default is YES.

### Yes

The default value. Specifies that the data mover is available for use.

### No

Specifies that the data mover is not available for use. When the hardware is being maintained, you can use the UPDATE DATAMOVER command to set the data mover off-line.

**Important:** If a library is controlled using a path from a NAS data mover to the library, and the NAS data mover is offline, the server will not be able to access the library. If the server is halted and restarted while the NAS data mover is offline, the library will not be initialized.

## DATAFormat (Required)

Specifies the data format that is used by this data mover.

### NETAPPDump

**NETAPPDUMP** must be used for NetApp NAS file servers and the IBM System Storage™ N Series.

### CELERRADump

**CELERRADUMP** must be used for EMC Celerra NAS file servers.

### NDMPDump

**NDMPDump** must be used for NAS file servers other than NetApp or EMC file servers.

## Example: Define a data mover for a NetApp file server by IP address

Define a data mover for the NAS node named NAS1. The numerical IP address for the data mover is 9.67.97.103, at port 10000. The NAS file server is a NetApp device.

```
define datamover nas1 type=nas haddress=9.67.97.103 laddress=10000 userid=root
password=admin dataformat=netappdump
```

## Example: Define a data mover by domain name

Define a data mover for the node named NAS1. The domain name for the data mover is, NETAPP2.TUCSON.IBM.COM at port 10000.

```
define datamover nas1 type=nas haddress=netapp2.tucson.ibm.com laddress=10000
userid=root password=admin dataformat=netappdump
```

## Example: Define a data mover by IP address

Define a data mover for the node named NAS1. The numerical IP address for the data mover is 9.67.97.103, at port 10000. The NAS file server is neither a NetApp or an EMC file server.

```
define datamover nas1 type=nas haddress=9.67.97.103 laddress=10000
userid=root password=admin dataformat=ndmpdump
```

## DEFINE DATAMOVER

### Related commands

*Table 66. Commands related to DEFINE DATAMOVER*

| Command          | Description  |
|------------------|--|
| DEFINE PATH      | Defines a path from a source to a destination.                 |
| DELETE DATAMOVER | Deletes a data mover.  |
| QUERY DATAMOVER  | Displays data mover definitions.                               |
| REGISTER NODE    | Defines a client to the server and sets options for that user. |
| UPDATE DATAMOVER | Changes the definition for a data mover.                       |



## DEFINE DEVCLASS (Define a device class)

Use this command to define a device class for a type of storage device. The server requires that a device class be defined to allow use of a device.

**Note:** The DISK device class is defined by IBM Tivoli Storage Manager and cannot be modified with the DEFINE DEVCLASS command.

The syntax and parameter descriptions are provided according to the device type. The syntax and parameter information is presented in the following order:

- 3570 (“Syntax” on page 140)
- 3590 (“Syntax” on page 143)
- 3592 (“Syntax” on page 147)
- 4MM (“Syntax” on page 152)
- 8MM (“Syntax” on page 156)
- CARTRIDGE (“Syntax” on page 162)
- CENTERA (“Syntax” on page 166)
- DLT (“Syntax” on page 168)
- DTF (“Syntax” on page 174)
- ECARTRIDGE (“Syntax” on page 177)
- FILE (“Syntax” on page 182)
- GENERICTAPE (“Syntax” on page 185)
- LTO (“Syntax” on page 188)
- NAS (“Syntax” on page 193)
- OPTICAL and WORM Types (“Syntax” on page 196)
- QIC (“Syntax” on page 200)
- REMOVABLEFILE (“Syntax” on page 205)
- SERVER (“Syntax” on page 208)
- VOLSAFE (“Syntax” on page 210)

*Table 67. Commands related to DEFINE DEVCLASS*

| Command          | Description   |
|------------------|---|
| BACKUP DEVCONFIG | Backs up IBM Tivoli Storage Manager device information to a file. |
| DEFINE LIBRARY   | Defines an automated or manual library.                           |
| DELETE DEVCLASS  | Deletes a device class name.                                      |
| QUERY DEVCLASS   | Displays information about device classes.                        |
| QUERY DIRSPACE   | Displays information about FILE directories.                      |
| UPDATE DEVCLASS  | Changes the attributes of a device class.                         |

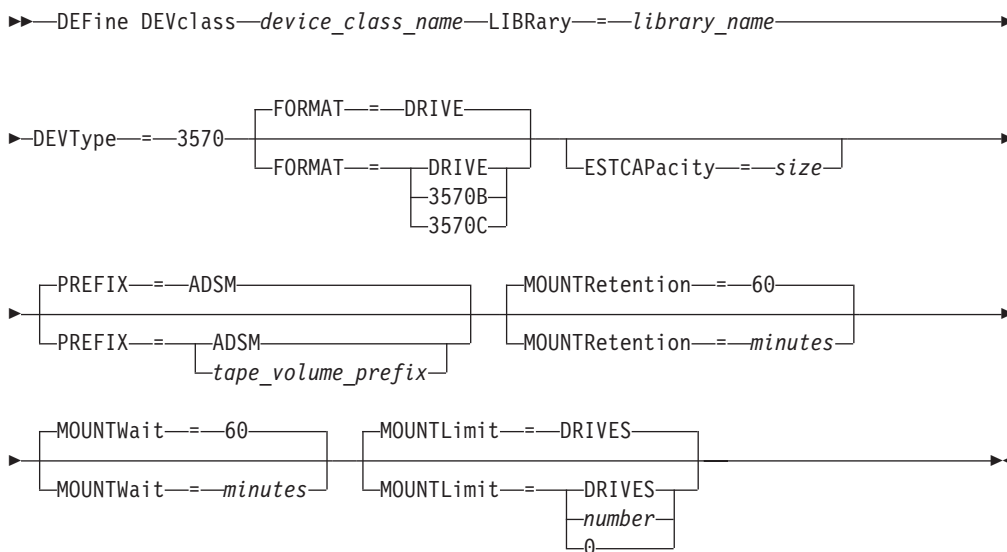
## DEFINE DEVCLASS (Update a 3570 device class)

Use the 3570 device class when you are using 3570 tape devices.

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

#### DEVType=3570 (Required)

Specifies the 3570 device type is assigned to the device class. The 3570 indicates that IBM 3570 cartridge tape devices are assigned to this device class.

#### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional. The default is `DRIVE`.

The following table lists the recording formats and estimated capacities for 3570 devices:

Table 68. Recording format and default estimated capacity for 3570 tape volumes

| Format | Estimated Capacity | Description  |
|--------|--------------------|--|
| DRIVE  | —                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3570B  | 5.0 GB             | Uncompressed (basic) format  |
| 3570C  | See note           | Compressed format  |
|        | 10.0 GB            |  |

**Note:** If 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.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for 3570 cartridge tapes, see Table 68.

### PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

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 a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

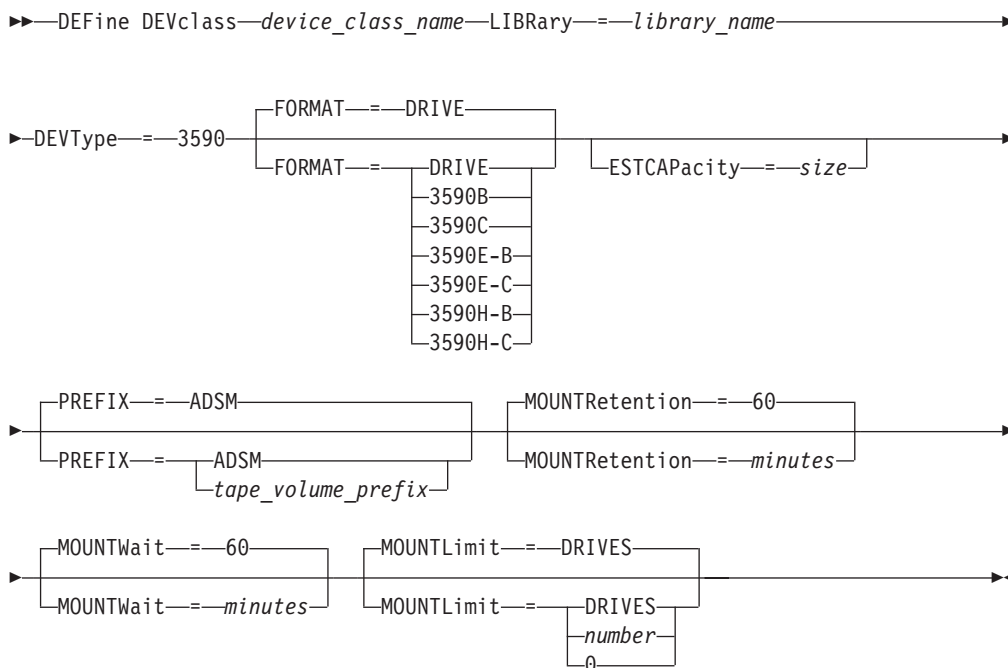
## DEFINE DEVCLASS (Define a 3590 device class)

Use the 3590 device class when you are using 3590 tape devices.

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

#### DEVType=3590 (Required)

Specifies the 3590 device type is assigned to the device class. 3590 indicates that IBM 3590 cartridge tape devices are assigned to this device class.

#### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional. The default value is `DRIVE`.

If the drives are in a library that includes drives of different tape technology, do not use the `DRIVE` value. Use the specific format that the drives use.

Refer to the *Administrator's Guide* for more information.

The following tables list the recording formats, estimated capacities and recording format options for 3590 devices:

*Table 69. Recording formats and default estimated capacities for 3590*

| Format  | Estimated Capacity  | Description  |
|---------|---|--|
| DRIVE   | –   | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3590B   | 10.0 GB   | Uncompressed (basic) format  |
| 3590C   | See note<br>20.0 GB   | Compressed format  |
| 3590E-B | 10.0 GB   | Uncompressed (basic) format, similar to the 3590B format   |
| 3590E-C | See note<br>20.0 GB   | Compressed format, similar to the 3590C format   |
| 3590H-B | 30.0 GB (J cartridge – standard – length)<br><br>60.0 GB (K cartridge - extended length)                | Uncompressed (basic) format, similar to the 3590B format   |
| 3590H-C | See note<br><br>60.0 GB (J cartridge - standard length)<br><br>120.0 GB (K cartridge - extended length) | Compressed format, similar to the 3590C format   |

**Note:** If 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.

*Table 70. 3590 device recording format selections*

| Device     | Format     |            |            |            |            |            |
|------------|------------|------------|------------|------------|------------|------------|
|            | 3590B      | 3590C      | 3590E-B    | 3590E-C    | 3590H-B    | 3590H-C    |
| 3590       | Read/Write | Read/Write | –          | –          | –          | –          |
| Ultra SCSI | Read/Write | Read/Write | –          | –          | –          | –          |
| 3590E      | Read       | Read       | Read/Write | Read/Write | –          | –          |
| 3590H      | Read       | Read       | Read       | Read       | Read/Write | Read/Write |

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

### **PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

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 a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.



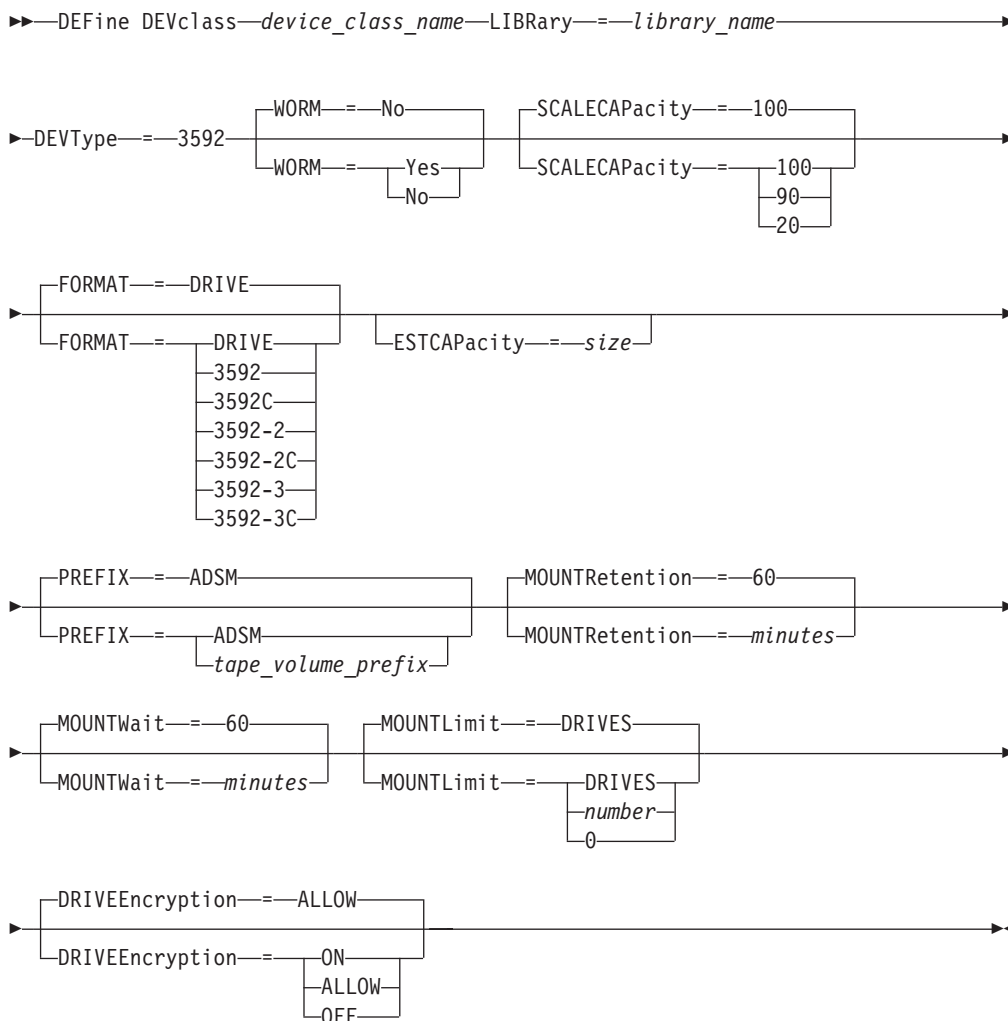
## DEFINE DEVCLASS (Define a 3592 device class)

Use the 3592 device class when you are using 3592 tape devices.

### Privilege class

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

### Syntax



### Parameters

#### device\_class\_name (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

#### DEVType=3592 (Required)

Specifies the 3592 device type is assigned to the device class.

### WORM

Specifies whether the drives use WORM (write once, read many) media. The parameter is optional. The default is NO.

#### Yes

Specifies that the drives will use WORM media.

#### No

Specifies that the drives will not use WORM media.

### Remember:

1. To use 3592 WORM support in 3584 libraries, you only need to specify the WORM parameter. The Tivoli Storage Manager server will distinguish between WORM and non-WORM scratch volumes. However, to use 3592 WORM support in 349X libraries, you also need to set the WORMSCRATCHCATEGORY on the DEFINE LIBRARY command. For details, see “DEFINE LIBRARY (Define a library)” on page 222.
2. When WORM=YES, the only valid value for the SCALECAPACITY parameter is 100.
3. Verify with your hardware vendors that your hardware is at the appropriate level of support.

For more information about WORM media, in general, refer to the *Administrator's Guide*.

### SCALECAPacity

Specifies the percentage of the media capacity that can be used to store data. This parameter is optional. The default is 100. Possible values are 20, 90, or 100.

Setting the scale capacity percentage to 100 provides maximum storage capacity. Setting it to 20 provides fastest access time.

**Note:** The scale capacity value will only take effect when data is first written to a volume. Any updates to the device class for scale capacity will not affect volumes that already have data written to them until the volume is returned to scratch status.

### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional. The default value is DRIVE.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats, estimated capacities and recording format options for 3592 devices:

Table 71. Recording formats and default estimated capacities for 3592

| Format  | Estimated Capacity | Description  |
|---------|--------------------|--|
| DRIVE   | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3592    | 300 GB             | Uncompressed (basic) format  |
| 3592C   | See note<br>900 GB | Compressed format  |
| 3592-2  | 500 GB<br>700 GB   | Uncompressed (basic) format JA tapes<br>Uncompressed (basic) format JB tapes   |
| 3592-2C | 1.5 TB<br>2.1 TB   | Compressed format JA tapes<br>Compressed format JB tapes   |
| 3592-3  | 640 GB<br>1 TB     | Uncompressed (basic) format JA tapes<br>Uncompressed (basic) format JB tapes   |
| 3592-3C | 1.9 TB<br>3 TB     | Compressed format JA tapes<br>Compressed format JB tapes   |

**Note:** If this format uses the tape drive hardware compression feature, depending on the effectiveness of compression, the actual capacity might be different than the listed value.

**Important:** For optimal performance, avoid mixing different generations of drives in a single SCSI library. If you must mix drive generations in an SCSI library, use one of the special configurations that are described in the *Administrator's Guide* to prevent or minimize media problems.

Special configurations are also required for mixing different generations of 3592 drives in 349x and ACSLS libraries.

For details, see the *Administrator's Guide*.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

### PREFIX

Specifies the high level qualifier of the data set name that the server writes into

the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

#### **MOUNTRetention**

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 a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

#### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

#### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Tip:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### **DRIVEEncryption**

Specifies whether drive encryption will be permitted. This parameter is optional. The default is ALLOW.

**ON**

Specifies that Tivoli Storage Manager is the key manager for drive encryption and will permit drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes—for example, backup sets, export volumes, and database backup volumes—will not be encrypted.) If you specify ON and you enable either the library or system method of encryption, drive encryption will not be permitted and backup operations will fail.

**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is permitted if either the library or system method of encryption is enabled.

**OFF**

Specifies that drive encryption will not be permitted. If you enable either the library or system method of encryption, backups will fail. If you enable the application method, Tivoli Storage Manager will disable encryption and backups will be attempted.

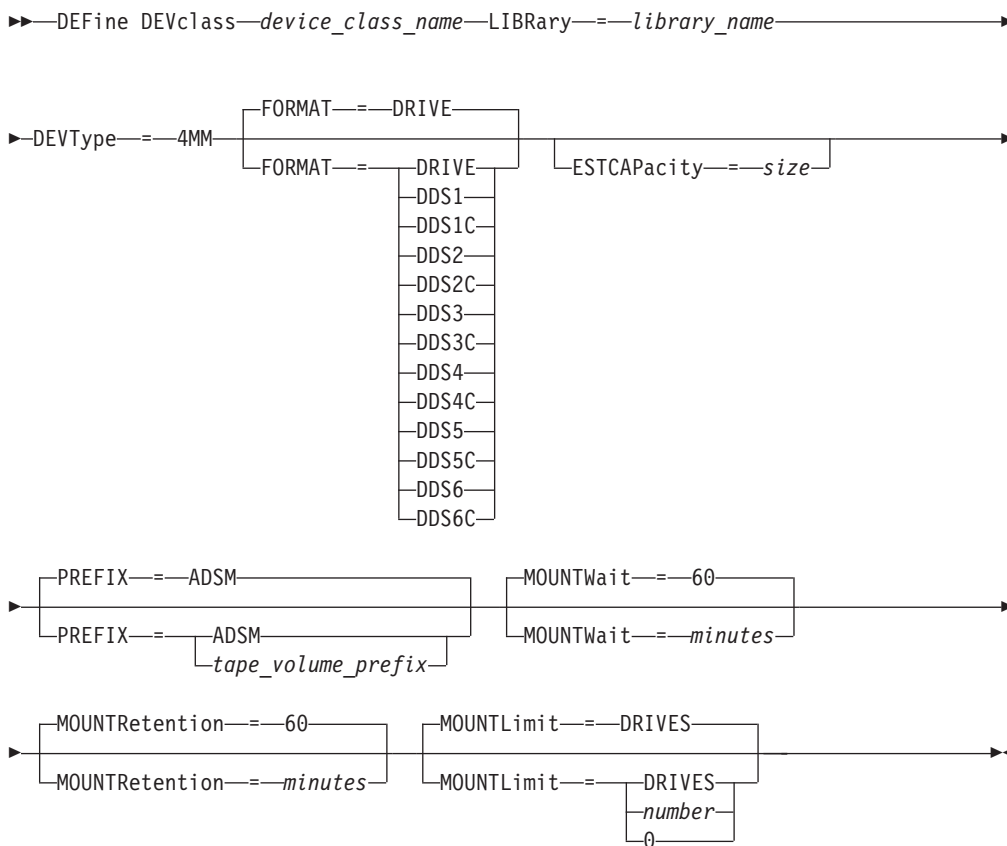
### DEFINE DEVCLASS (Define a 4MM device class)

Use the 4MM device class when you are using 4 mm tape devices.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### LIBRARY (Required)

Specifies the name of the defined library object that contains the 4 mm tape drives used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

##### DEVType=4MM (Required)

Specifies that the 4MM device type is assigned to the device class. The 4MM indicates that 4 mm tape devices are assigned to this device class.

##### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional. The default value is `DRIVE`.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use.

Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats and estimated capacities for 4 mm devices:

Table 72. Recording formats and default estimated capacities for 4 mm tapes

| Format | Estimated Capacity                                 | Description  |
|--------|--|--|
| DRIVE  | —  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| DDS1   | 2.6 GB (60-meter)<br>4.0 GB (90-meter)             | Uncompressed format, only applies to 60-meter and 90-meter tapes   |
| DDS1C  | See note<br>1.3 GB (60-meter)<br>2.0 GB (90-meter) | Compressed format, only applies to 60-meter and 90-meter tapes   |
| DDS2   | 4.0 GB   | Uncompressed format, only applies to 120-meter tapes   |
| DDS2C  | See note<br>8.0 GB                                 | Compressed format, only applies to 120-meter tapes   |
| DDS3   | 12.0 GB  | Uncompressed format, only applies to 125-meter tapes   |
| DDS3C  | See note<br>24.0 GB                                | Compressed format, only applies to 125-meter tapes   |
| DDS4   | 20.0 GB  | Uncompressed format, only applies to 150-meter tapes   |
| DDS4C  | See note<br>40.0 GB                                | Compressed format, only applies to 150-meter tapes   |
| DDS5   | 36 GB  | Uncompressed format, when using DAT 72 media   |
| DDS5C  | See note<br>72 GB                                  | Compressed format, when using DAT 72 media   |
| DDS6   | 80 GB  | Uncompressed format, when using DAT 160 media  |
| DDS6C  | See note<br>160 GB                                 | Compressed format, when using DAT 160 media  |

**Note:** If 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.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for 4 mm tapes, see Table 72 on page 153.

### PREFIX

Specifies the high level qualifier of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### MOUNTLimit

Specifies the maximum number of sequential access volumes that can be



simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

#### 0 (zero)

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

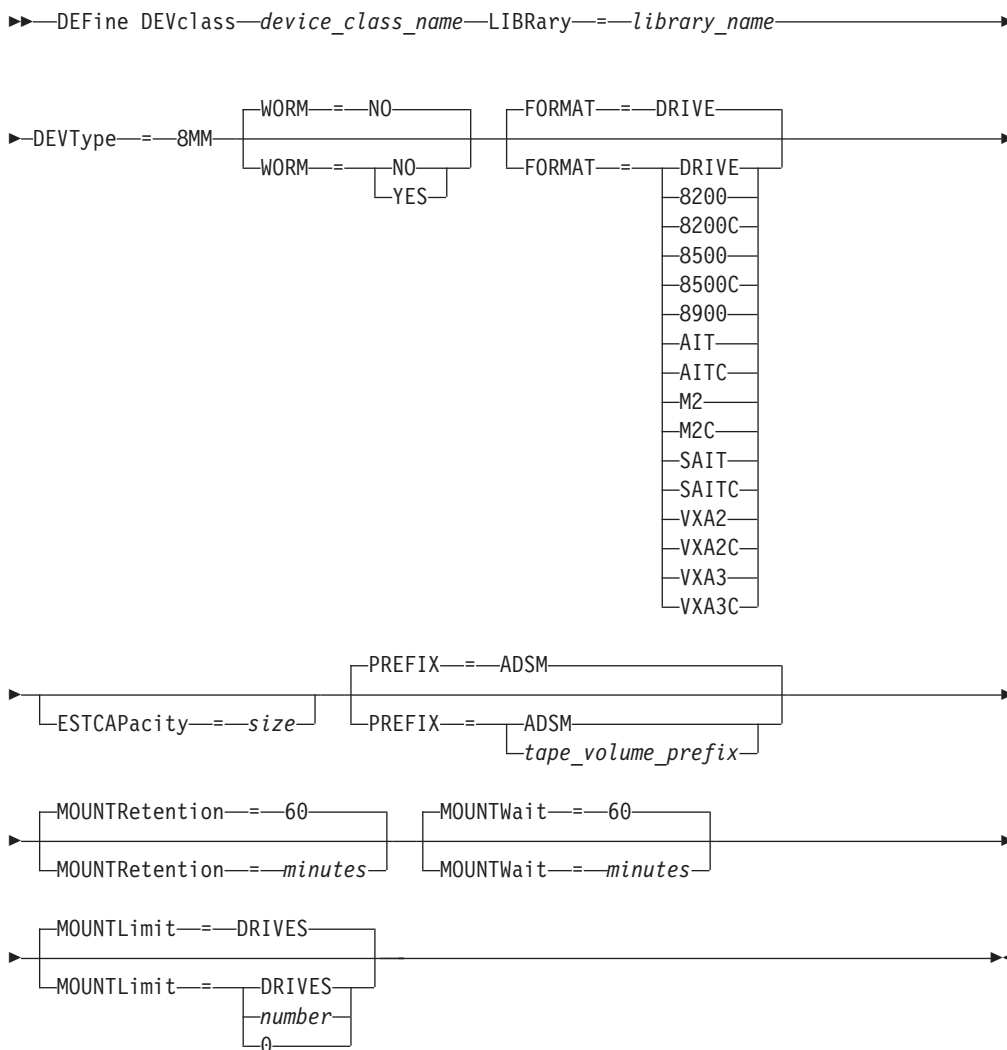
## DEFINE DEVCLASS (Define an 8MM device class)

Use the 8MM device class when you are using 8 mm tape devices.

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the 8 mm tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### DEVType=8MM (Required)

Specifies that the 8MM device type is assigned to the device class. 8MM indicates that 8 mm tape devices are assigned to this device class.

### WORM

Specifies whether the drives will use WORM (write once, read many) media. The parameter is optional. The default is NO.

#### Yes

Specifies that the drives will use WORM media.

**Note:** If you select YES, the only options available for the FORMAT parameter are:

- DRIVE
- AIT
- AITC

#### No

Specifies that the drives will not use WORM media.

Refer to the *Administrator's Guide* for more information.

### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional. The default value is DRIVE.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats and estimated capacities for 8 mm devices:

Table 73. Recording format and default estimated capacity for 8 mm tape

| Format      |                              |  |
|-------------|------------------------------|--|
| Medium Type | Estimated Capacity           | Description  |
| DRIVE       | —                            | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 8200        | 2.3 GB                       | Uncompressed (standard) format, using standard 112-meter tape cartridges   |
| 8200C       | See note<br>3.5 GB<br>4.6 GB | Compressed format, using standard 112-meter tape cartridges  |

*Table 73. Recording format and default estimated capacity for 8 mm tape (continued)*

| <b>Format</b>      |                           |                                |
|--------------------|---------------------------|--------------------------------|
| <b>Medium Type</b> | <b>Estimated Capacity</b> | <b>Description</b>             |
| 8500               | See note                  | Drives (Read Write)            |
| 15m                | 600 MB                    | Eliant 820 (RW)                |
| 15m                | 600 MB                    | Exabyte 8500/8500C (RW)        |
| 15m                | 600 MB                    | Exabyte 8505 (RW)              |
| 54m                | 2.35 GB                   | Eliant 820 (RW)                |
| 54m                | 2.35 GB                   | Exabyte 8500/8500C (RW)        |
| 54m                | 2.35 GB                   | Exabyte 8505 (RW)              |
| 112m               | 5 GB or 10.0 GB           | Eliant 820 (RW)                |
| 112m               | 5 GB or 10.0 GB           | Exabyte 8500/8500C (RW)        |
| 112m               | 5 GB or 10.0 GB           | Exabyte 8505 (RW)              |
| 160m XL            | 7 GB                      | Eliant 820 (RW)                |
| 8500C              | See note                  | Drives (Read Write)            |
| 15m                | 1.2 GB                    | Eliant 820 (RW)                |
| 15m                | 1.2 GB                    | Exabyte 8500/8500C (RW)        |
| 15m                | 1.2 GB                    | Exabyte 8505 (RW)              |
| 54m                | 4.7 GB                    | Eliant 820 (RW)                |
| 54m                | 4.7 GB                    | Exabyte 8500/8500C (RW)        |
| 54m                | 4.7 GB                    | Exabyte 8505 (RW)              |
| 112m               | 5 GB or 10.0 GB           | Eliant 820 (RW)                |
| 112m               | 5 GB or 10.0 GB           | Exabyte 8500/8500C (RW)        |
| 112m               | 5 GB or 10.0 GB           | Exabyte 8505 (RW)              |
| 160m XL            | 7 GB                      | Eliant 820 (RW)                |
| 8900               | See note                  | Drive (Read Write)             |
| 15m                | –                         | Mammoth 8900 (R)               |
| 54m                | –                         | Mammoth 8900 (R)               |
| 112m               | –                         | Mammoth 8900 (R)               |
| 160m XL            | –                         | Mammoth 8900 (R)               |
| 22m                | 2.5 GB                    | Mammoth 8900 (RW)              |
| 125m               | –                         | Mammoth 8900 (RW with upgrade) |
| 170m               | 40 GB                     | Mammoth 8900 (RW)              |
| AIT                | See note                  | Drive                          |
| SDX1–25C           | 25 GB                     | AIT, AIT2 and AIT3 drives      |
| SDX1–35C           | 35 GB                     | AIT, AIT2 and AIT3 drives      |
| SDX2–36C           | 36 GB                     | AIT2 and AIT3 drives           |
| SDX2–50C           | 50 GB                     | AIT2 and AIT3 drives           |
| SDX3–100C          | 100 GB                    | AIT3, AIT4, and AIT5 drives    |
| SDX3X–150C         | 150 GB                    | AIT3-Ex, AIT4, and AIT5 drives |
| SDX4–200C          | 200 GB                    | AIT4 and AIT5 drives           |
| SDX5–400C          | 400 GB                    | AIT5 drive                     |
| AITC               | See note                  | Drive                          |
| SDX1–25C           | 50 GB                     | AIT, AIT2 and AIT3 drives      |
| SDX1–35C           | 91 GB                     | AIT, AIT2 and AIT3 drives      |
| SDX2–36C           | 72 GB                     | AIT2 and AIT3 drives           |
| SDX2–50C           | 130 GB                    | AIT2 and AIT3 drives           |
| SDX3–100C          | 260 GB                    | AIT3, AIT4, and AIT5 drives    |
| SDX3X–150C         | 390 GB                    | AIT3-Ex, AIT4, and AIT5 drives |
| SDX4–200C          | 520 GB                    | AIT4 and AIT5 drives           |
| SDX5–400C          | 1040 GB                   | AIT5 drive                     |

Table 73. Recording format and default estimated capacity for 8 mm tape (continued)

| Format      |                    |                    |
|-------------|--------------------|--------------------|
| Medium Type | Estimated Capacity | Description        |
| M2          | See note           | Drive (Read Write) |
| 75m         | 20.0 GB            | Mammoth II (RW)    |
| 150m        | 40.0 GB            | Mammoth II (RW)    |
| 225m        | 60.0 GB            | Mammoth II (RW)    |
| M2C         | See note           | Drive (Read Write) |
| 75m         | 50.0 GB            | Mammoth II (RW)    |
| 150m        | 100.0 GB           | Mammoth II (RW)    |
| 225m        | 150.0 GB           | Mammoth II (RW)    |
| SAIT        | See note           | Drive (Read Write) |
|             | 500 GB             | Sony SAIT1-500(RW) |
| SAITC       | See note           | Drive (Read Write) |
|             | 1300 GB (1.3 TB)   | Sony SAIT1-500(RW) |
| VXA2        | See note           | Drive (Read Write) |
| V6 (62m)    | 20 GB              | VXA-2              |
| V10 (124m)  | 40 GB              |                    |
| V17 (170m)  | 60 GB              |                    |
| VXA2C       | See note           | Drive (Read Write) |
| V6 (62m)    | 40 GB              | VXA-2              |
| V10 (124m)  | 80 GB              |                    |
| V17 (170m)  | 120 GB             |                    |
| VXA3        | See note           | Drive (Read Write) |
| X6 (62m)    | 40 GB              | VXA-3              |
| X10 (124m)  | 86 GB              |                    |
| X23 (230m)  | 160 GB             |                    |
| VXA3C       | See note           | Drive (Read Write) |
| X6 (62m)    | 80 GB              | VXA-3              |
| X10 (124m)  | 172 GB             |                    |
| X23 (230m)  | 320 GB             |                    |

**Note:** The actual capacities may vary depending on which cartridges and drives are used.

- For the M2C format, the normal compression ratio is 2.5:1.
- For the AITC and SAITC formats, the normal compression ratio is 2.6:1.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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.

Specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB. The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for 8 mm tapes, see Table 73 on page 157.

### **PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

For EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

The following are possible values:

## DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

## **Example: Define an 8 mm device class**

Define a device class named 8MMTAPE for an 8 mm 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 6 GB.

```
define devclass 8mmtape devtype=8mm library=auto
format=drive mountlimit=2 mountretention=10
prefix=adsmvol estcapacity=6G
```

## DEFINE DEVCLASS — CARTRIDGE

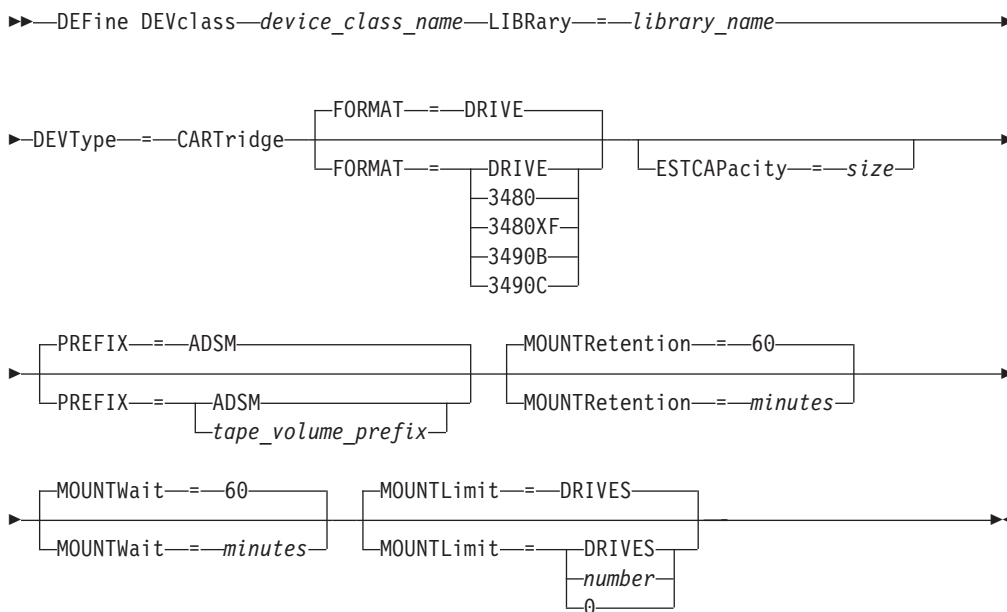
### DEFINE DEVCLASS (Define a cartridge device class)

Use the CARTRIDGE device class when you are using IBM 3480 and 3490 cartridge tape devices.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### LIBRARY (Required)

Specifies the name of the defined library object that contains the CARTRIDGE tape drives that can be used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

##### DEVType=CARTRIDGE (Required)

Specifies that the CARTRIDGE device type is assigned to the device class. CARTRIDGE indicates that cartridge tape devices are assigned to this device class.

##### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use.

Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats and estimated capacities for Cartridge System Tapes (CST):



Table 74. Recording format and default estimated capacity for cartridge tape volumes

| Format | Estimated capacity | Description  |
|--------|--------------------|--|
| DRIVE  | —                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3480   | 180 MB             | 18-track basic recording format (CST)  |
| 3480XF | 180 MB             | 18-track compressed recording format (CST)   |
| 3490B  | See note           | 36-track   |
|        | 360 MB<br>720 MB   | Uncompressed (basic) recording format (CST)<br>Compressed recording format (ECCST)   |
| 3490C  | See note           | 36-track   |
|        | 360 MB<br>720 MB   | Uncompressed (basic) recording format (CST)<br>Compressed recording format (ECCST)   |

**Note:** The actual capacities may vary depending on which cartridges and drives are used.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for cartridge tapes, see Table 74.

### PREFIX

Specifies the high-level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional. The default value is ADSM.

If you have already established a tape label naming convention that supports your current tape management system, use a tape volume prefix that conforms to your naming conventions.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.

- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

ADSM.BFS is an example of a tape volume filename using the default prefix and the added server qualifier.

### **MOUNTRetention**

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 a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

**Example: Define a cartridge device class**

Define a device class named `SQUARE1` with a `CARTRIDGE` tape device and specify 30 minutes as the amount of time to retain idle mounted volumes for the device class.

```
define devclass square1 devtype=cartridge mountretention=30
```

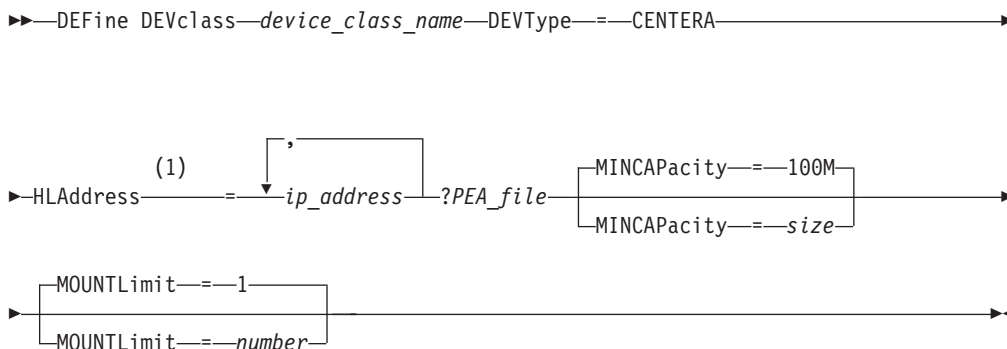
### DEFINE DEVCLASS (Define a Centera device class)

Use the CENTERA device class when you are using EMC Centera storage devices. The CENTERA device type uses files as volumes to store data sequentially. It is similar to the FILE device class.

#### Privilege class

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

#### Syntax



#### Notes:

- For each Centera device class, you must specify one or more IP addresses. However, a Pool Entry Authorization (PEA) file name and path are optional, and up to one PEA file specification can follow the IP addresses. Use the "?" character to separate the PEA file name and path from the IP addresses.

#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### DEVType=CENTERA (Required)

Specifies that the Centera device type is assigned to this device class. All volumes belonging to a storage pool that is defined to this device class are logical volumes that are a form of sequential access media

##### HLAddress

Specifies one or more IP addresses for the Centera storage device and, optionally, the name and path of one Pool Entry Authorization (PEA) file. Specify the IP addresses using the dotted decimal format (for example, 9.10.111.222). A Centera device might have multiple IP addresses. If multiple IP addresses are specified, then the store or retrieve operation will attempt a connection using each IP address specified until a valid address is found.

The PEA file name and path name are case sensitive.

If you append the name and path of a PEA file, ensure that the file is stored in a directory on the system running the Tivoli Storage Manager server. Separate the PEA file name and path from the IP address using the "?" character, for example:

```
HLADDRESS=9.10.111.222,9.10.111.223?/user/ControlFiles/TSM.PEA
```

Specify only one PEA file name and path for each device class definition. If you specify two different Centera device classes that point to the same Centera storage device and if the device class definitions contain different PEA file names and paths, the Tivoli Storage Manager server will use the PEA file specified in the device class HLADDRESS parameter that was first used to open the Centera storage device.

### Tips:

1. The Tivoli Storage Manager server does *not* include a PEA file during installation. If you do not create a PEA file, the Tivoli Storage Manager server uses the Centera default profile, which can allow applications to read, write, delete, purge, and query data on a Centera storage device. To provide tighter control, create a PEA file using the command line interface provided by EMC Centera. For details about Centera authentication and authorization, refer to the EMC Centera *Programmer's Guide*.
2. You can also specify the PEA file name and path in an environment variable using the syntax `CENTERA_PEA_LOCATION=filePath_fileName`. The PEA file name and path specified using this environment variable apply to all Centera clusters. If you use this variable, you do not need to specify the PEA file name and path using the HLADDRESS parameter.

### MINCAPacity

Specifies the minimum size for Centera volumes assigned to a storage pool in this device class. This value represents the minimum amount of data stored on a Centera volume before the Tivoli Storage Manager server marks it full. Centera volumes will continue to accept data until the minimum amount of data has been stored. This parameter is optional.

Specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The default value is 100 MB (MINCAPACITY=100M). The minimum value allowed is 1 MB (MINCAPACITY=1M). The maximum value allowed is 128 GB (MINCAPACITY=128G).

### MOUNTLimit

Specifies the maximum number of files that can be simultaneously open for input and output. The default value is 1. This parameter is optional. You can specify any number from 0 or greater; however, the sum of all mount limit values for all device classes assigned to the same Centera device should not exceed the maximum number of sessions allowed by Centera.

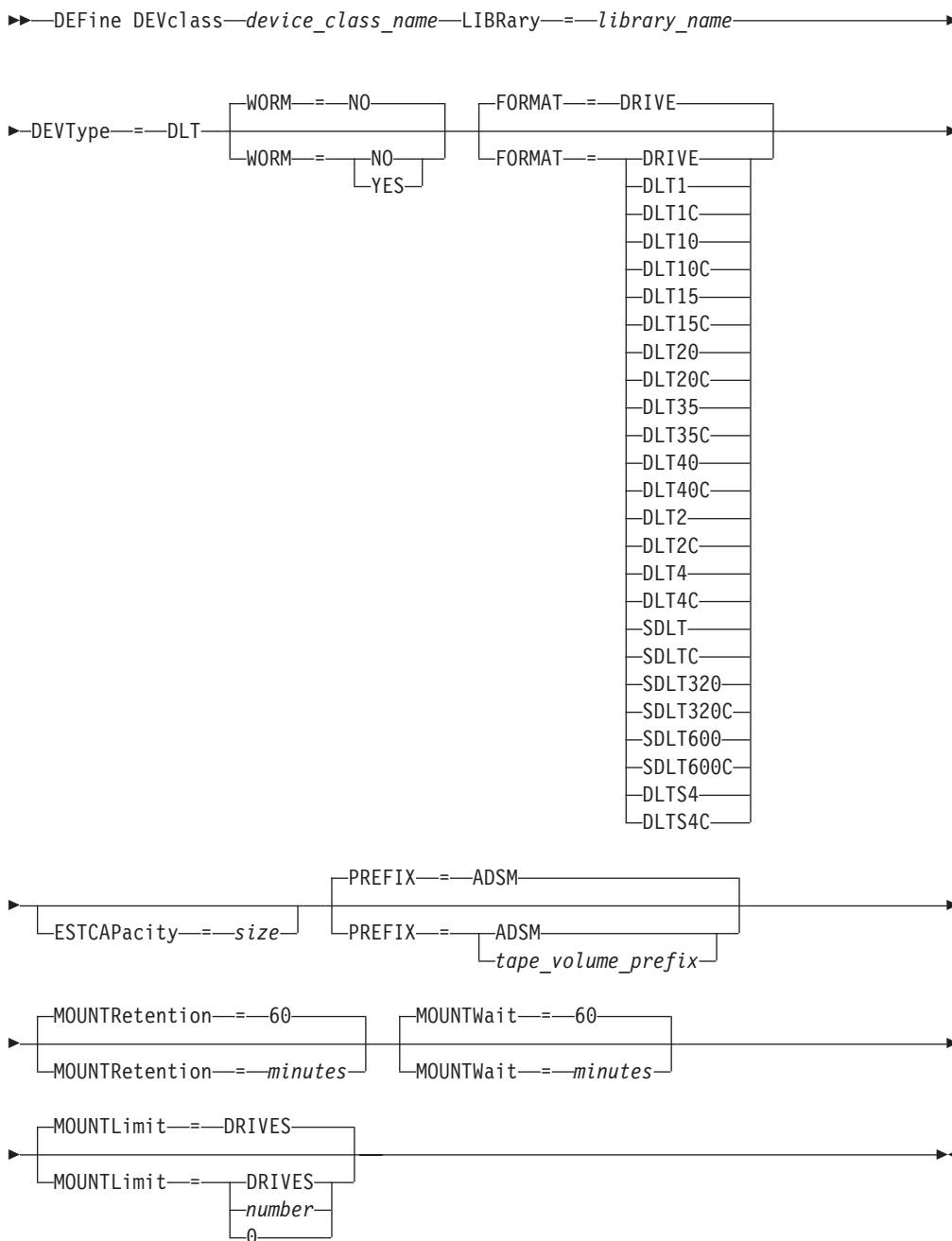
### DEFINE DEVCLASS (Define a DLT device class)

Use the DLT device class when you are using DLT tape devices.

#### Privilege class

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

#### Syntax



## Parameters

### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

### LIBRARY (Required)

Specifies the name of the defined library object that contains the DLT tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### DEVType=DLT (Required)

Specifies that the DLT device type is assigned to the device class. DLT indicates that DLT tape devices are assigned to this device class.

### WORM

Specifies whether the drives will use WORM (write once, read many) media. The parameter is optional. The default is NO.

#### Yes

Specifies that the drives will use WORM media.

**Note:** Support for DLT WORM media is available only for SDLT-600, Quantum DLT-V4, and Quantum DLT-S4 drives in manual, SCSI, and ACSLS libraries.

#### No

Specifies that the drives will not use WORM media.

For more information, refer to the *Administrator's Guide*.

### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats and estimated capacities for DLT devices:

Table 75. Recording format and default estimated capacity for DLT

| Format | Estimated Capacity | Description   |
|--------|--------------------|---|
| DRIVE  | –                  | <p>The server selects the highest format that is supported by the drive on which a volume is mounted.</p> <p><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives.</p> |
| DLT1   | 40.0 GB            | <p>Uncompressed format, using only CompacTape III cartridges</p> <p>Valid with DLT4000, DLT7000, and DLT8000 drives</p>   |

*Table 75. Recording format and default estimated capacity for DLT (continued)*

| <b>Format</b> | <b>Estimated Capacity</b> | <b>Description</b>  |
|---------------|---------------------------|---|
| DLT1C         | See note 1 on page 171.   | Compressed format, using only CompacTape III cartridges     |
|               | 80.0 GB                   | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT10         | 10.0 GB                   | Uncompressed format, using only CompacTape III cartridges   |
|               |                           | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT10C        | See note 1 on page 171.   | Compressed format, using only CompacTape III cartridges     |
|               | 20.0 GB                   | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT15         | 15.0 GB                   | Uncompressed format, using only CompacTape IIIxt cartridges |
|               |                           | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT15C        | See note 1 on page 171.   | Compressed format, using only CompacTape IIIxt cartridges   |
|               | 30.0 GB                   | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT20         | 20.0 GB                   | Uncompressed format, using only CompacTape IV cartridges    |
|               |                           | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT20C        | See note 1 on page 171.   | Compressed format, using only CompacTape IV cartridges      |
|               | 40.0 GB                   | Valid with DLT4000, DLT7000, and DLT8000 drives             |
| DLT35         | 35.0 GB                   | Uncompressed format, using only CompacTape IV cartridges    |
|               |                           | Valid with DLT7000 and DLT8000 drives                       |
| DLT35C        | See note 1 on page 171.   | Compressed format, using only CompacTape IV cartridges      |
|               | 70.0 GB                   | Valid with DLT7000 and DLT8000 drives                       |
| DLT40         | 40.0 GB                   | Uncompressed format, using CompacTape IV cartridges         |
|               |                           | Valid with a DLT8000 drive                                  |
| DLT40C        | See note 1 on page 171.   | Compressed format, using CompacTape IV cartridges           |
|               | 80.0 GB                   | Valid with a DLT8000 drive                                  |
| DLT2          | 80.0 GB                   | Uncompressed format, using Quantum DLT tape VS1 media       |
| DLT2C         | See note 1 on page 171.   | Compressed format, using Quantum DLT tape VS1 media         |
|               | 160.0 GB                  |   |



*Table 75. Recording format and default estimated capacity for DLT (continued)*

| <b>Format</b>           | <b>Estimated Capacity</b> | <b>Description</b>  |
|-------------------------|---------------------------|---|
| DLT4                    | 160.0 GB                  | Uncompressed format, using Quantum DLTtape VS1 cartridges.<br><br>Valid with Quantum DLT-V4 drive |
| DLT4C                   | See note 1.<br>320.0 GB   | Compressed format, using Quantum DLTtape VS1 cartridges.<br><br>Valid with Quantum DLT-V4 drive   |
| SDLT<br>See note 2.     | 100.0 GB                  | Uncompressed format, using Super DLT Tape 1 cartridges<br><br>Valid with a Super DLT drive        |
| SDLTC<br>See note 2.    | See note 1.<br>200.0 GB   | Compressed format, using Super DLT Tape 1 cartridges<br><br>Valid with a Super DLT drive          |
| SDLT320<br>See note 2.  | 160.0 GB                  | Uncompressed format, using Quantum SDLT I media<br><br>Valid with a Super DLT drive               |
| SDLT320C<br>See note 2. | See note 1.<br>320.0 GB   | Compressed format, using Quantum SDLT I media<br><br>Valid with a Super DLT drive                 |
| SDLT600                 | 300.0 GB                  | Uncompressed format, using SuperDLTtape-II media<br><br>Valid with a Super DLT drive              |
| SDLT600C                | See note 1.<br>600.0 GB   | Compressed format, using SuperDLTtape-II media<br><br>Valid with a Super DLT drive                |
| DLTS4                   | 800 GB                    | Uncompressed format, using Quantum DLT S4 media.<br><br>Valid with a DLT-S4 drive                 |
| DLTS4C                  | See note 1.<br>1.6 TB     | Compressed format, using Quantum DLT S4 media.<br><br>Valid with a DLT-S4 drive                   |

**Note:**

1. Depending on the effectiveness of compression, the actual capacity may be greater than the listed value.
2. Tivoli Storage Manager does not support a library that contains both Backward Read Compatible (BRC) SDLT and Non-Backward Read Compatible (NBRC) SDLT drives.

**ESTCAPacity**

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB. The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For more information on estimated capacities, see Table 75 on page 169.

### **PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

## DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

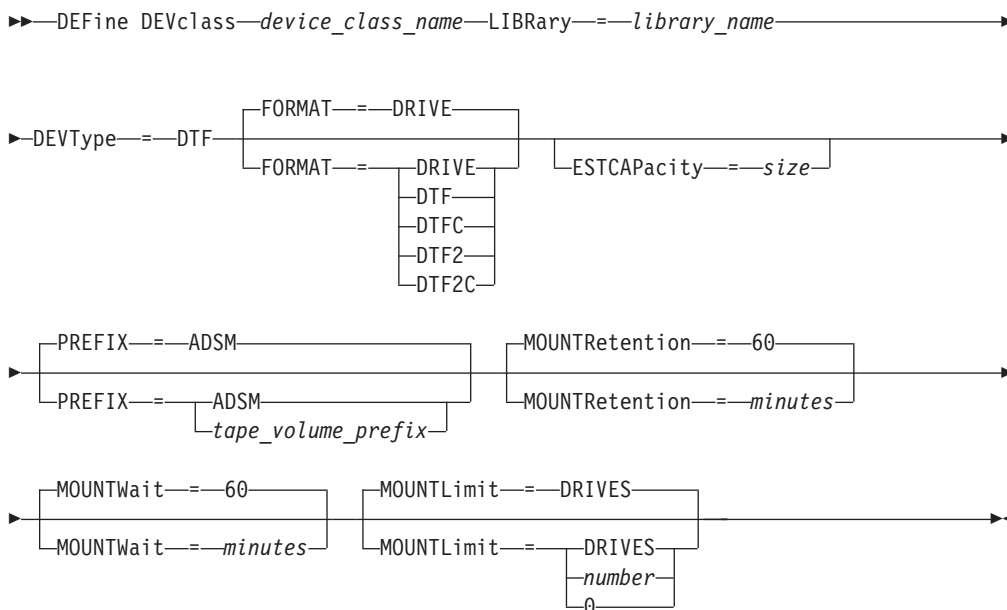
### DEFINE DEVCLASS (Define a DTF device class)

Use the DTF device class when you are using DTF tape devices.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### LIBRARY (Required)

Specifies the name of the defined library object that contains the DTF tape drives used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

##### DEVType=DTF (Required)

Specifies that the DTF tape devices are assigned to this device class.

##### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is `DRIVE`.

If the drives are in a library that includes drives of different tape technology, do not use the `DRIVE` value. Use the specific format that the drives use.

Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats and estimated capacities for DTF devices:

Table 76. Recording format and default estimated capacity for DTF

| Format | Estimated Capacity   | Description  |
|--------|----------------------|--|
| DRIVE  | –                    | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| DTF    | 12.0 GB<br>42.0 GB   | Using GW-240 tape cassettes<br>Using GW-730L tape cassettes  |
| DTFC   | 24.0 GB<br>84.0 GB   | Using GW-240 tape cassettes<br>Using GW-730L tape cassettes  |
| DTF2   | 60.0 GB<br>200.0 GB  | Using GW2-60GS tape cassettes<br>Using GW2-200GL tape cassettes  |
| DTF2C  | 120.0 GB<br>400.0 GB | Using GW2-60GS tape cassettes<br>Using GW2-200GL tape cassettes  |

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For more information on estimated capacities, see Table 76.

### PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

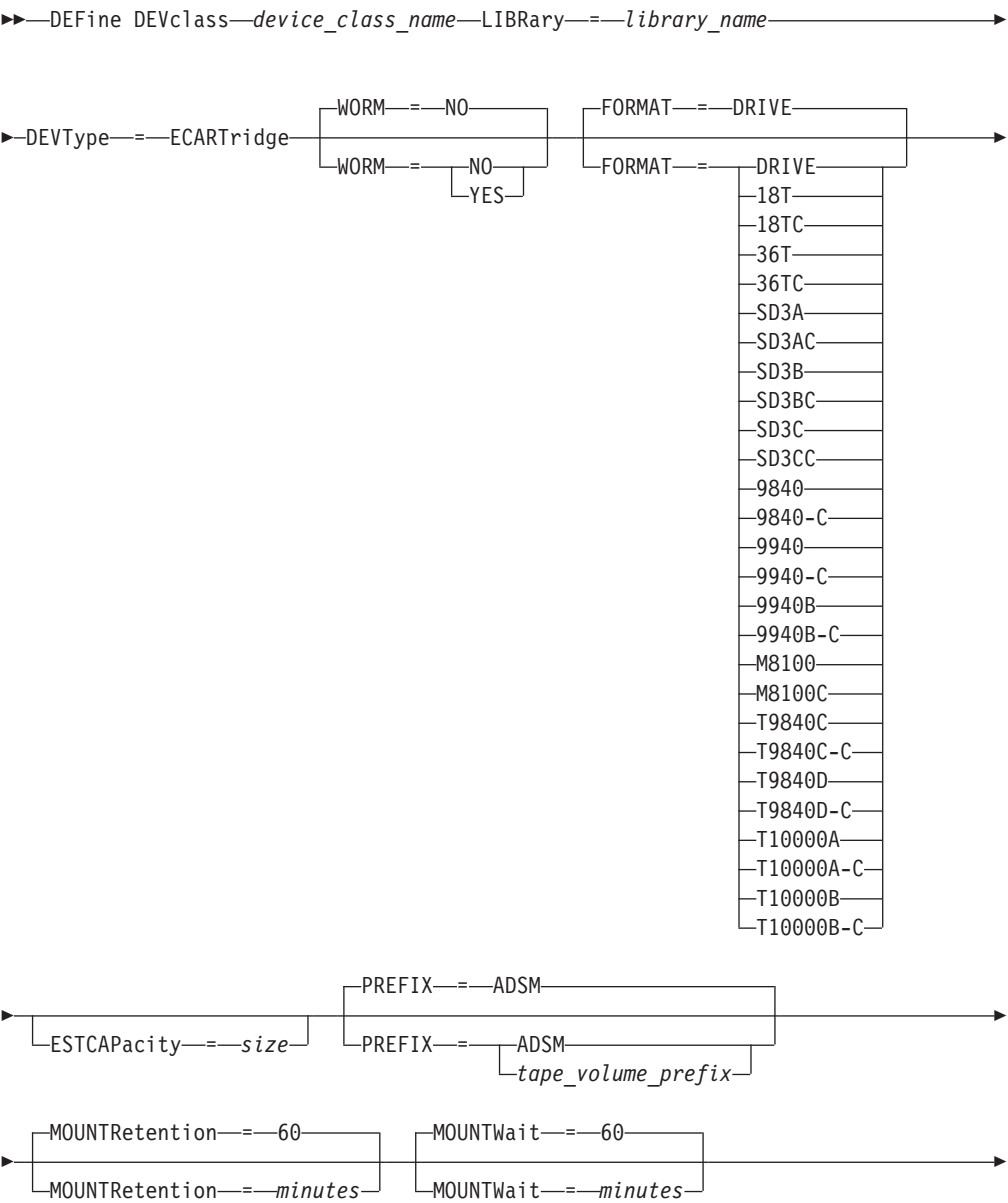
DEFINE DEVCLASS (Define an ECARTRIDGE device class)

Use the ECARTRIDGE device class when you are using StorageTek drives such as the StorageTek SD-3, 9490, 9840 or 9940.

Privilege class

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

Syntax



## DEFINE DEVCLASS — ECARTRIDGE



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the ECARTRIDGE tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

#### DEVType=ECARTridge (Required)

Specifies that the ECARTRIDGE device type is assigned to the device class. ECARTRIDGE indicates that a specific type of cartridge tape device (StorageTek) is assigned to this device class.

#### WORM

Specifies whether the drives use WORM (write once, read many) media. The parameter is optional. The default is NO.

**Restriction:** If you select YES, the only options that are available for the FORMAT parameter are:

- DRIVE
- 9840
- 9840-C
- T9840D
- T9840D-C
- T10000A
- T10000A-C
- T10000B
- T10000B-C

Refer to the *Administrator's Guide* for more information.

#### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE. If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to the *Administrator's Guide* for more information.

**Important:** If you specify DRIVE for a device class that has non-compatible sequential access devices, then you must mount volumes on devices that are capable of reading or writing the format established when the volume was first mounted. This can cause delays if the only sequential access device that can access the volume is already in use.

The following table lists the recording formats and estimated capacities for ECARTRIDGE devices:



Table 77. Recording formats and default estimated capacities for ECARTRIDGE tapes

| Format  | Estimated capacity | Description  |
|---------|--------------------|--|
| DRIVE   | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 18T     | 360 MB             | 18-track uncompressed (standard) (read-only) format  |
| 18TC    | 1.44 GB            | 18-track extended (read-only), compressed format   |
| 36T     | 720 MB             | 36-track extended (read and write) format  |
| 36TC    | 2.88 GB            | 36-track extended (read and write), compressed format  |
| SD3A    | 10 GB              | Uncompressed (standard) format, using a 10 GB 'A' cartridge with 91 meters (298 feet) of tape  |
| SD3AC   | 40 GB              | Compressed format, using a 10 GB 'A' cartridge with 91 meters (298 feet) of tape   |
| SD3B    | 25 GB              | Uncompressed (standard) format, using a 25 GB 'B' cartridge with 204 meters (668 feet) of tape   |
| SD3BC   | 100 GB             | Compressed format, using a 25 GB 'B' cartridge with 204 meters (668 feet) of tape  |
| SD3C    | 50 GB              | Uncompressed (standard) format, using a 50 GB 'C' cartridge with 392 meters (1286 feet) of tape  |
| SD3CC   | 200 GB             | Compressed format, using a 50 GB 'C' cartridge with 392 meters (1286 feet) of tape   |
| 9840    | 20 GB              | Uncompressed 9840 format, using a Sun StorageTek 9840 cartridge  |
| 9840-C  | 80 GB              | LZ-1 Enhanced (4:1) compressed 9840 format, using a Sun StorageTek 9840 cartridge  |
| 9940    | 60 GB              | Uncompressed 9940 format, using a Sun StorageTek 9940 cartridge  |
| 9940-C  | 120 GB             | Compressed 9940 format, using a Sun StorageTek 9940 cartridge  |
| 9940B   | 200 GB             | Uncompressed 9940B format, using a Sun StorageTek 9940 cartridge   |
| 9940B-C | 400 GB             | Compressed 9940B format, using a Sun StorageTek 9940 cartridge   |
| M8100   | 10 GB              | Uncompressed (standard) format, using a 10 GB cartridge  |
| M8100C  | 10 GB              | Compressed format, using a 10 GB cartridge. Because there is no mechanism to determine the type of cartridge in an M8100 drive, the server assumes a 10 GB estimated uncompressed capacity.  |
| T9840C  | 40 GB              | Uncompressed T9840C format, using a Sun StorageTek 9840 cartridge  |

## DEFINE DEVCLASS — ECARTRIDGE

Table 77. Recording formats and default estimated capacities for ECARTRIDGE tapes (continued)

| Format    | Estimated capacity | Description  |
|-----------|--------------------|--|
| T9840C-C  | 80 GB              | Compressed T9840C format, using a Sun StorageTek 9840 cartridge      |
| T9840D    | 75 GB              | Uncompressed T9840D format, using a Sun StorageTek 9840 cartridge    |
| T9840D-C  | 150 GB             | Compressed T9840D format, using a Sun StorageTek 9840 cartridge      |
| T10000A   | 500 GB             | Uncompressed T10000A format, using a Sun StorageTek T10000 cartridge |
| T10000A-C | 1 TB               | Compressed T10000A format, using a Sun StorageTek T10000 cartridge   |
| T10000B   | 1 TB               | Uncompressed T10000B format, using a Sun StorageTek T10000 cartridge |
| T10000B-C | 2 TB               | Compressed T10000B format, using a Sun StorageTek T10000 cartridge   |

**Notes:**

- Some formats use a tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity may be double or more than the listed value.
- T10000A drives can read and write the T10000A format only. T10000B drives can read, but cannot write, the T10000A format.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

### PREFIX

Specifies the high-level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional. The default value is ADSM.

If you have already established a tape label naming convention that supports your current tape management system, use a tape volume prefix that conforms to your naming conventions.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.

- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

ADSM.BFS is an example of a tape volume filename using the default prefix and the added server qualifier.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Tip:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### DEFINE DEVCLASS (Define a FILE device class)

Use the FILE device class when you are using files on magnetic disk storage as volumes that store data sequentially (as on tape).

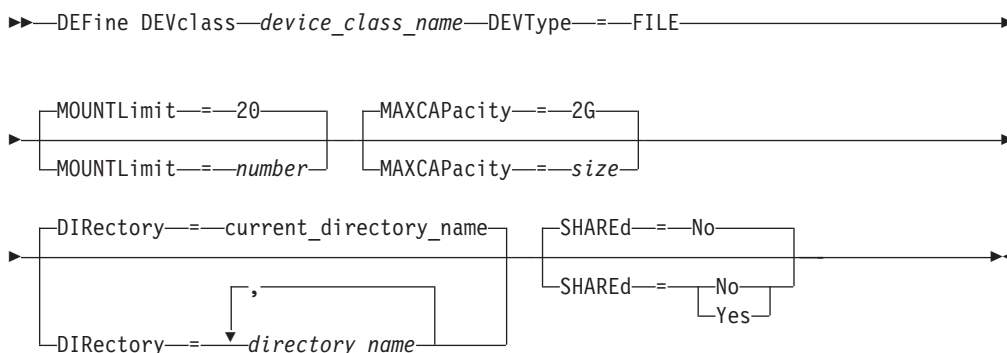
For information about disk subsystem requirements for FILE-type disk storage, see the *Administrator's Guide*.

The FILE device class does not support EXTERNAL libraries.

### Privilege class

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

### Syntax



### Parameters

#### device\_class\_name (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### DEVType=FILE (Required)

Specifies that the FILE device type is assigned to the device class. FILE indicates that a file is 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

Specifies the maximum number of files that can be simultaneously open for input and output. This parameter is optional. The default value is 20. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

#### MAXCAPacity

Specifies the maximum size of any data storage files defined to a storage pool in this device class.

The value of the MAXCAPACITY parameter is also used as the unit of allocation when storage pool space triggers create volumes. The default value is 2 GB (MAXCAPACITY=2G). The value specified should be less than or equal to the maximum supported size of a file on the target file system.

Specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The minimum value you can use is 100 KB (MAXCAPACITY=100K).

Do not define a MAXCAPACITY value greater than 640M when this file is for REMOVABLEFILE CD support. A value less than a CD's usable space (650 MB) enables a one-to-one match between files from the FILE device class and copies that are on CD.

### DIRECTORY

Specifies the directory location or locations of the files used in this device class. Enclose the entire list of directories within quotation marks, using commas to separate individual directory names. Special characters (for example, blank spaces) are permitted within directory names. For example, the directory list "abc def,xyz" contains two directories: abc def and xyz.

This parameter is optional.

The default is the current working directory of the server at the time the command is issued.

By specifying a directory name or names, you identify the location where the server places the files that represent storage volumes for this device class.

For NetApp SnapLock support (storage pools with RECLAMATIONTYPE=SNAPLOCK, which are going to use this device class), the directory or directories specified with DIRECTORY parameter must point to the directory or directories on the NetApp SnapLock volumes. For a detailed description of NetApp SnapLock support, refer to the *Administrator's Guide*.

While processing the command, the server expands the specified directory name or names into their fully qualified forms, starting from the root directory.

If the server needs to allocate a scratch volume, it creates a new file in one of these directories. (The server can choose any of the directories in which to create new scratch volumes.) 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, if you define a device class with a directory of tsmstor and the server needs a scratch volume in this device class to store export data, the file that the server creates might be named /tsmstor/00566497.exp.

**Important:** You must ensure that storage agents can access newly created FILE volumes. Failure of the storage agent to access a FILE volume can cause operations to be retried on a LAN-only path or to fail. For more information, see the description of the DIRECTORY parameter in "DEFINE PATH (Define a path)" on page 240.

**Tip:** If you specify multiple directories for a device class, ensure that the directories are associated with separate file systems. Space trigger functions and storage pool space calculations take into account the space remaining in each directory. If you specify multiple directories for a device class and the directories reside in the same file system, the server will calculate space by adding values representing the space remaining in each directory. These space calculations will be inaccurate. Rather than choosing a storage pool with

sufficient space for an operation, the server might choose the wrong storage pool and run out of space prematurely. For space triggers, an inaccurate calculation might result in a failure to expand the space available in a storage pool. Failure to expand space in a storage pool is one of the conditions that can cause a trigger to become disabled. If a trigger is disabled because the space in a storage pool could not be expanded, you can re-enable the trigger by issuing the following command: `update spacetrigger stg`. No further changes are required to the space trigger.

### SHARED

Specifies that this FILE device class will be shared between the server and one or more storage agents. To prepare for sharing, a library will be automatically defined along with a number of drives corresponding to the MOUNTLIMIT parameter value. The drive names are the name of the library plus a number from 1 to the mount limit number. For example, if the library name is FILE and the mount limit is set to 4, the drives are named FILE11, FILE12, FILE13, FILE14.

For information about prerequisites when storage is shared by the server and storage agent, see <http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html>.

### Example: Define a FILE device class with multiple directories

Define a device class that specifies multiple directories.

```
define devclass multidir devtype=file
    directory=/usr/xyz,/usr/abc,/usr/uvw
```

### Example: Define a FILE device class with a 50 MB capacity

Define a device class named PLAINFILES with a FILE device type and a maximum capacity of 50 MB.

```
define devclass plainfiles devtype=file
maxcapacity=50m
```

## DEFINE DEVCLASS (Define a GENERICTAPE device class)

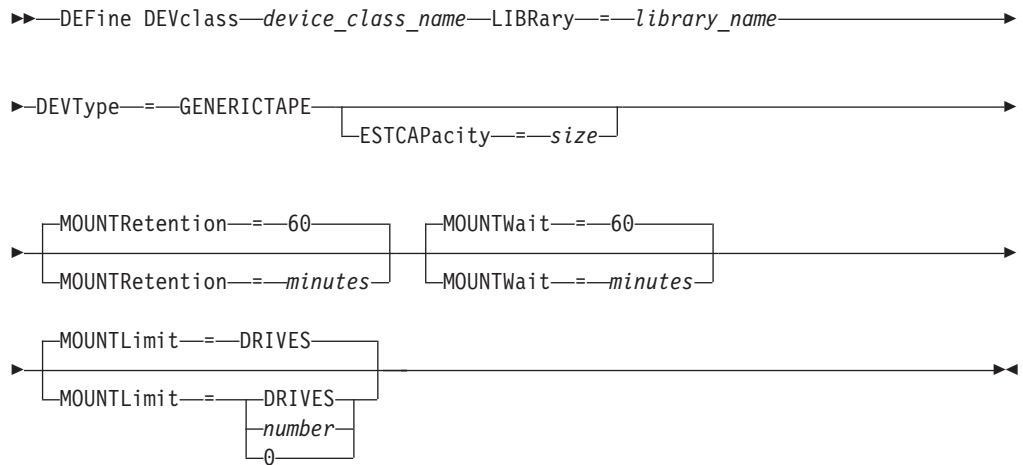
Use the GENERICTAPE device class for tape drives supported by operating system device drivers.

When using this device type, the server does not recognize either the type of device or the cartridge recording format. Because the server does not recognize the type of device, if an I/O error occurs, error information is less detailed compared to error information for a specific device type (for example, 8MM). When defining devices to the server, do not mix various types of devices within the same device type.

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the tape drives used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

#### DEVType=GENERICTAPE (Required)

Specifies that the GENERICTAPE device type is assigned to the device class. GENERICTAPE indicates that the volumes for this device class are used in tape drives supported by the operating system's tape device driver.

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 GENERICTAPE are sequential access volumes.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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.

Specify a capacity appropriate to the particular tape drive being used.

You must specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

### MOUNTRetention

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 a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### MOUNTLimit

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device



class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

## DEFINE DEVCLASS (Define an LTO device class)

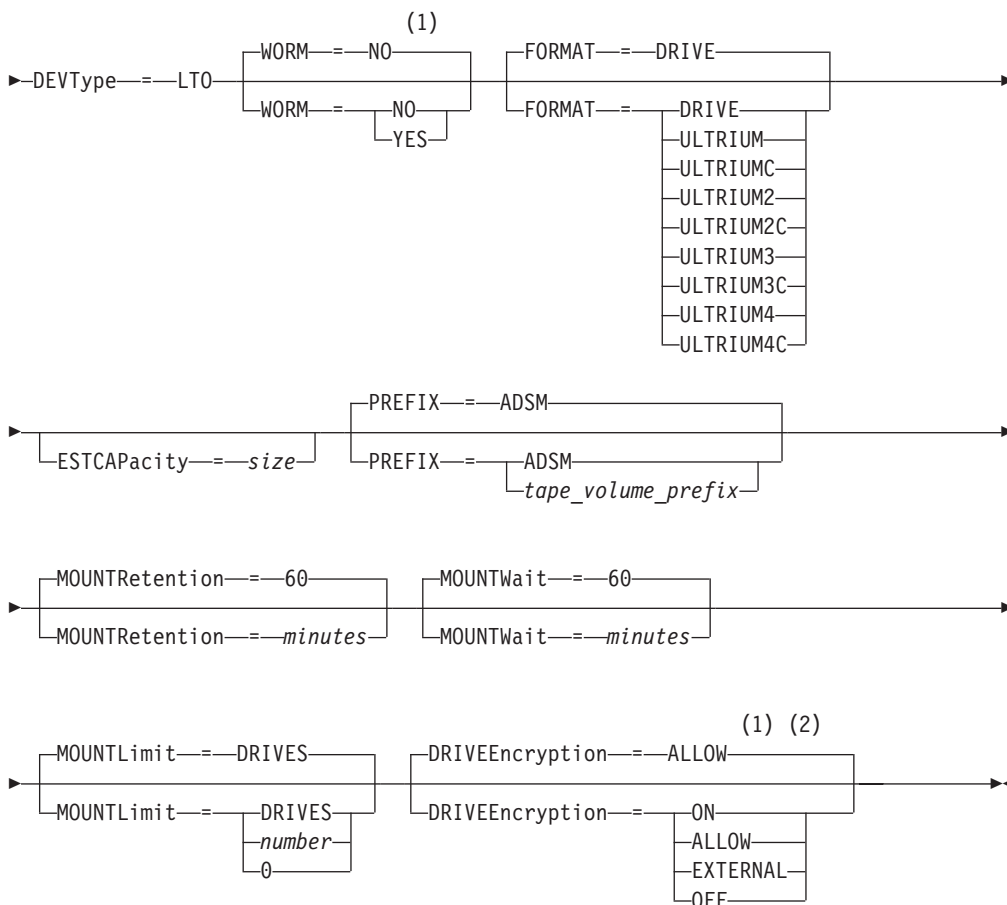
Use the LTO device class when you are using LTO tape devices.

### Privilege class

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

### Syntax

►► DEFINE DEVclass *device\_class\_name* LIBRARY=*library\_name* ►►



### Notes:

- 1 You cannot specify both WORM=YES and DRIVEENCRIPTION=ON.
- 2 Drive encryption is supported only for IBM and HP Ultrium 4 drives and media.

### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

### LIBRARY (Required)

Specifies the name of the defined library object that contains the LTO tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### DEVType=LTO (Required)

Specifies that the linear tape open (LTO) device type is assigned to the device class.

### WORM

Specifies whether the drives will use WORM (write once, read many) media. The parameter is optional. The default is NO.

#### Yes

Specifies that the drives will use WORM media.

#### Note:

1. To use WORM media in a library, all the drives in the library must be WORM capable.
2. You cannot specify Tivoli Storage Manager as the key manager for drive encryption of WORM (write once, read many) media. (Specifying both WORM=YES and DRIVEENCRYPTION=ON is not supported.)

#### No

Specifies that the drives will not use WORM media.

For more information, refer to the *Administrator's Guide*.

### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to the *Administrator's Guide* for more information.

When migrating all drives from Ultrium to Ultrium 2 devices:

- Delete all existing Ultrium drive definitions and the paths associated with them.
- Define the new Ultrium 2 drives and paths.

If you are considering mixing different generations of LTO media and drives, be aware of the following restrictions. For more information about mixing drives and media, refer to the *Administrator's Guide*.

Table 78. Read - write capabilities for different generations of LTO drives

| Drives                    | Generation 1 media | Generation 2 media | Generation 3 media | Generation 4 media |
|---------------------------|--------------------|--------------------|--------------------|--------------------|
| Generation 1              | Read and write     | n/a                | n/a                | n/a                |
| Generation 2              | Read and write     | Read and write     | n/a                | n/a                |
| Generation 3 <sup>1</sup> | Read only          | Read and write     | Read and write     | n/a                |
| Generation 4 <sup>2</sup> | n/a                | Read only          | Read and write     | Read and write     |

<sup>1</sup> In a library with a Generation 3 drive, all Generation 1 scratch volumes need to be checked out, and all Generation 1 storage pool volumes need to be updated to read-only.

<sup>2</sup> In a library with a Generation 4 drive, all Generation 2 scratch volumes need to be checked out, and all Generation 2 storage pool volumes need to be updated to read-only.

The following table lists the recording formats and estimated capacities for LTO devices:

*Table 79. Recording format and default estimated capacity for LTO*

| Format    | Estimated capacity | Description  |
|-----------|--------------------|--|
| DRIVE     | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| ULTRIUM   | 100 GB             | Uncompressed format, using Ultrium cartridges  |
| ULTRIUMC  | See note<br>200 GB | Compressed format, using Ultrium cartridges  |
| ULTRIUM2  | 200 GB             | Uncompressed (standard) format, using Ultrium 2 cartridges   |
| ULTRIUM2C | See note<br>400 GB | Compressed format, using Ultrium 2 cartridges  |
| ULTRIUM3  | 400 GB             | Uncompressed (standard) format, using Ultrium 3 cartridges   |
| ULTRIUM3C | See note<br>800 GB | Compressed format, using Ultrium 3 cartridges  |
| ULTRIUM4  | 800 GB             | Uncompressed (standard) format, using Ultrium 4 cartridges   |
| ULTRIUM4C | See note<br>1.6 TB | Compressed format, using Ultrium 4 cartridges  |

**Note:** If 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.

## ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=100G specifies that the estimated capacity for a volume in this device class is 100 GB.

For more information on estimated capacities, see Table 79.

## PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned

to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, you can enhance device sharing between applications by setting this parameter to a low value (for example, two minutes).

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### **DRIVEEncryption**

Specifies whether drive encryption will be permitted. This parameter is optional. The default is ALLOW.

#### **Restriction:**

1. Drive encryption is supported only for IBM and HP Ultrium 4 drives and media.
2. You cannot specify Tivoli Storage Manager as the key manager for drive encryption of WORM (write once, read many) media. (Specifying both WORM=YES and DRIVEENCRYPTION=ON is not supported.)

### **ON**

Specifies that Tivoli Storage Manager is the key manager for drive encryption and will permit drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes will not be encrypted. For example, backup sets, export volumes, and database backup volumes are not encrypted.) If you specify ON and you enable another method of encryption, drive encryption will not be permitted and backup operations will fail.

### **ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is permitted if another method of encryption is enabled.

### **EXTERNAL**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. Use this setting with an encryption methodology that is provided by another vendor and that is used with Application Method Encryption (AME) enabled on the drive. When you specify EXTERNAL and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager does not turn encryption off. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns encryption off.

### **OFF**

Specifies that drive encryption will not be permitted. If you enable another method of encryption, backups will fail. If you enable the application method, Tivoli Storage Manager will disable encryption and backups will be attempted.

### **Example: Define an LTO device class**

Define a device class named LTOTAPE for an LTO drive in a library named LTOLIB. The format is ULTRIUM, mount limit is 12, mount retention is 5, tape volume prefix is named SMVOL, and the estimated capacity is 100 GB.

```
define devclass ltotape devtype=lto library=ltolib
format=ultrium mountlimit=12 mountretention=5
prefix=smvol estcapacity=100G
```

## DEFINE DEVCLASS (Define a NAS device class)

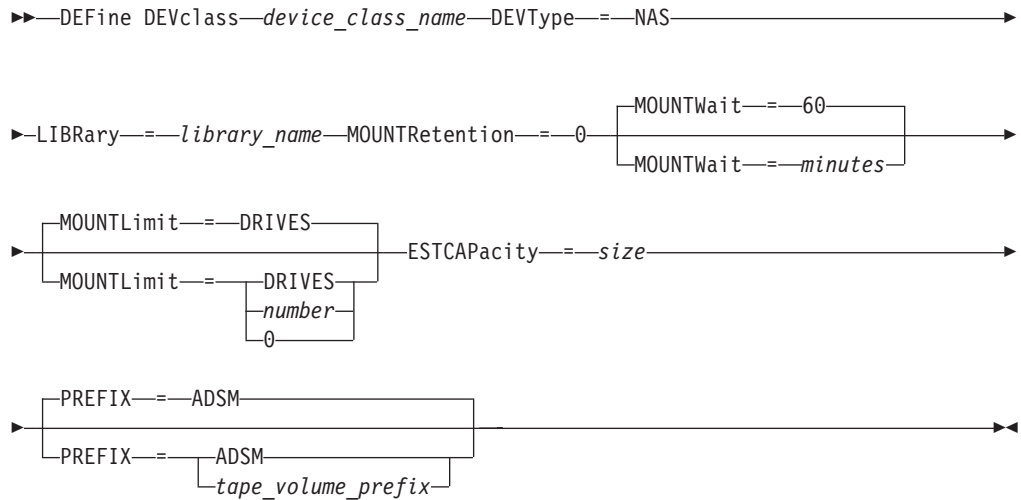
Use the NAS device class when you are using NDMP (Network Data Management Protocol) operations to back up network-attached storage (NAS) file servers. The device class is for drives supported by the NAS file server for backups.

The NAS device class does not support EXTERNAL libraries.

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### DEVType=NAS (Required)

Specifies that the network-attached storage (NAS) device type is assigned to the device class. The NAS device type is for drives attached to and used by a NAS file server for backup of NAS file systems.

#### LIBRARY (Required)

Specifies the name of the defined library object that contains the SCSI tape drives used by this device class. For information about defining a library object, see the `DEFINE LIBRARY` command.

#### MOUNTRetention=0 (Required)

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. Zero (0) is the only supported value for device classes with `DEVType=NAS`.

#### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Tip:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### **ESTCAPacity (Required)**

Specifies the estimated capacity for the volumes assigned to this device class.

You must specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=100G specifies that the estimated capacity for a volume in this device class is 100 GB.

### **PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.



- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is  
ADSM.BFS.

**Example: Define a NAS device class**

Define a device class named NASTAPE for a NAS drive in a library named NASLIB. The mount limit is DRIVES, mount retention is 0, tape volume prefix is named SMVOL, and the estimated capacity is 200 GB.

```
define devclass nastape devtype=nas library=naslib  
mountretention=0 mountlimit=drives  
prefix=smvol estcapacity=200G
```

## DEFINE DEVCLASS — OPTICAL and WORM types

### DEFINE DEVCLASS (Define OPTICAL and WORM device classes)

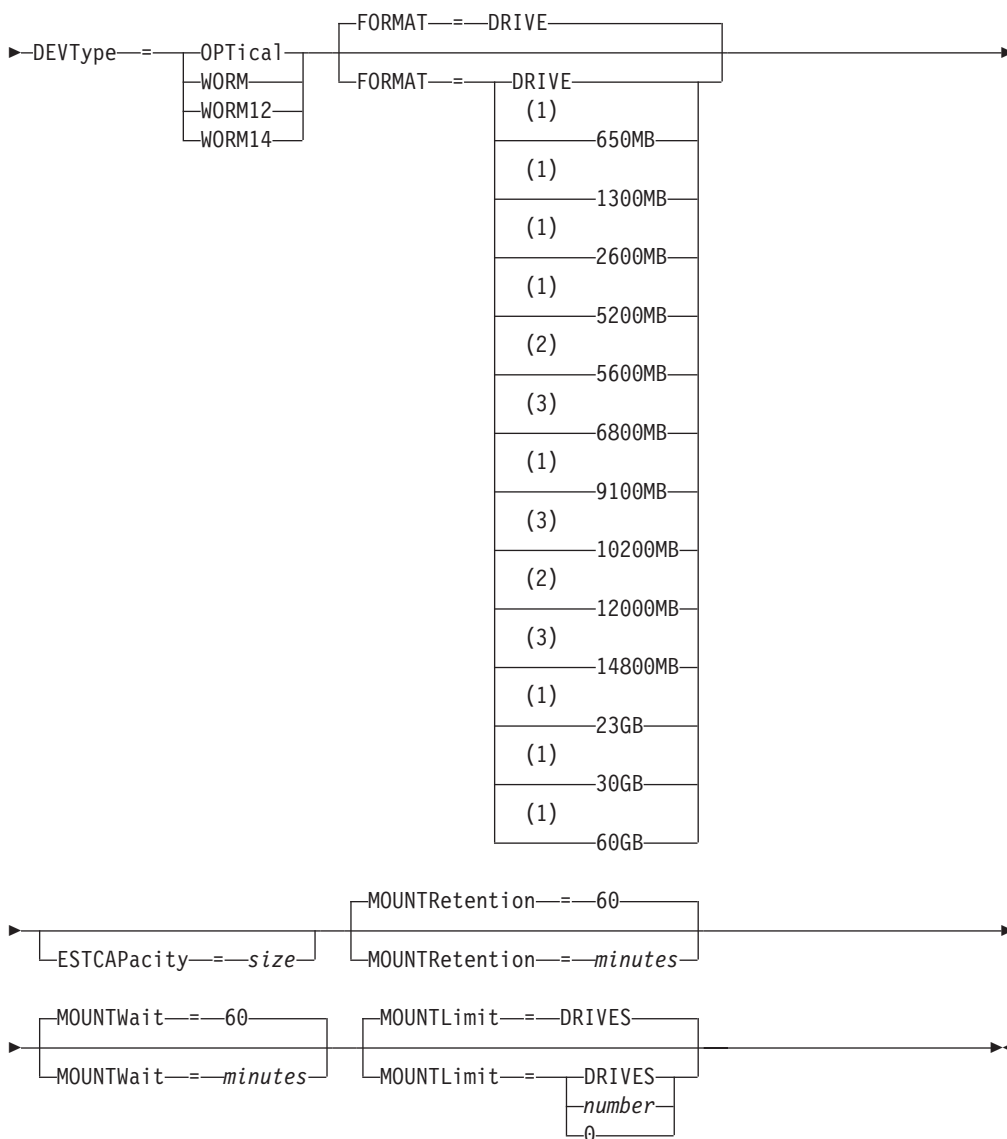
Use the OPTICAL device class when you are using optical devices. Use the WORM device class when you are using optical devices that have WORM capability.

#### Privilege class

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

#### Syntax

►►—DEFine DEVclass—*device\_class\_name*—LIBRary—==—*library\_name*—————►



#### Notes:

- 1 Not available for WORM12 or WORM14 device types
- 2 Available only for WORM12 device types

3 Available only for WORM14 device types

## Parameters

### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

### LIBRARY (Required)

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.

### DEVType (Required)

Specifies the device types that are assigned to the device class.

Possible values are:

#### OPTical

Specifies that the device class uses two-sided 5.25 inch rewritable optical media.

#### WORM

Specifies that the device class uses two-sided 5.25 inch write-once read-many (WORM) optical media.

Do not use this device type for 12-inch or 14-inch WORM optical media.

#### WORM12

Specifies that the device class uses one-sided 12-inch WORM optical media.

#### WORM14

Specifies that the device class uses two-sided 14-inch WORM optical media.

## FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE. If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats and estimated capacities for OPTICAL and WORM devices:

Table 80. Estimated capacities for OPTICAL and WORM drives

| Format | Estimated capacity | Description  |
|--------|--------------------|--|
| DRIVE  | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 650MB  | 650 MB             | Using a 650 MB 5.25-inch optical drive<br><br>Not valid for WORM12 or WORM14 device types  |

## DEFINE DEVCLASS — OPTICAL and WORM types

Table 80. Estimated capacities for OPTICAL and WORM drives (continued)

| Format  | Estimated capacity | Description  |
|---------|--------------------|--|
| 1300MB  | 1300 MB            | Using a 1300 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types             |
| 2600MB  | 2600 MB            | Using a 2600 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types             |
| 5200MB  | 5200 MB            | Using a 5200 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types             |
| 5600MB  | 5600 MB            | Using a 5600 MB 12-inch optical drive<br>Valid for WORM12 device types only                        |
| 6800MB  | 6800 MB            | Using a 6800 MB 14-inch optical drive<br>Valid for WORM14 device types only                        |
| 9100MB  | 9100 MB            | Using a 9100 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types             |
| 10200MB | 10200 MB           | Using a 10200 MB 14-inch optical drive<br>Only valid for WORM14 device types                       |
| 12000MB | 12000 MB           | Using a 12000 MB 12-inch optical drive<br>Valid for WORM12 device types only                       |
| 14800MB | 14800 MB           | Using a 14800 MB 14-inch optical drive<br>Valid for WORM14 device types only                       |
| 23GB    | 23 GB              | Using Sony Blue Laser read-write and WORM media<br>Not valid for WORM12 or WORM14 device types     |
| 30GB    | 30 GB              | Using Plasmon UDO1 read-write and WORM media<br>Not valid for WORM12 or WORM14 device types        |
| 60GB    | 60 GB              | Using Plasmon or IBM UDO2 read-write and WORM media<br>Not valid for WORM12 or WORM14 device types |

**Restriction:** If you are considering mixing different generations of UDO media and drives, be aware of the following:

- UDO1 drives can read and write UDO1 media only.
- UDO2 drives can read from, but not write to, UDO1 media.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on estimated capacities, see Table 80 on page 197.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true mount limit value (including online status).

**Tip:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

### DEFINE DEVCLASS (Define a QIC device class)

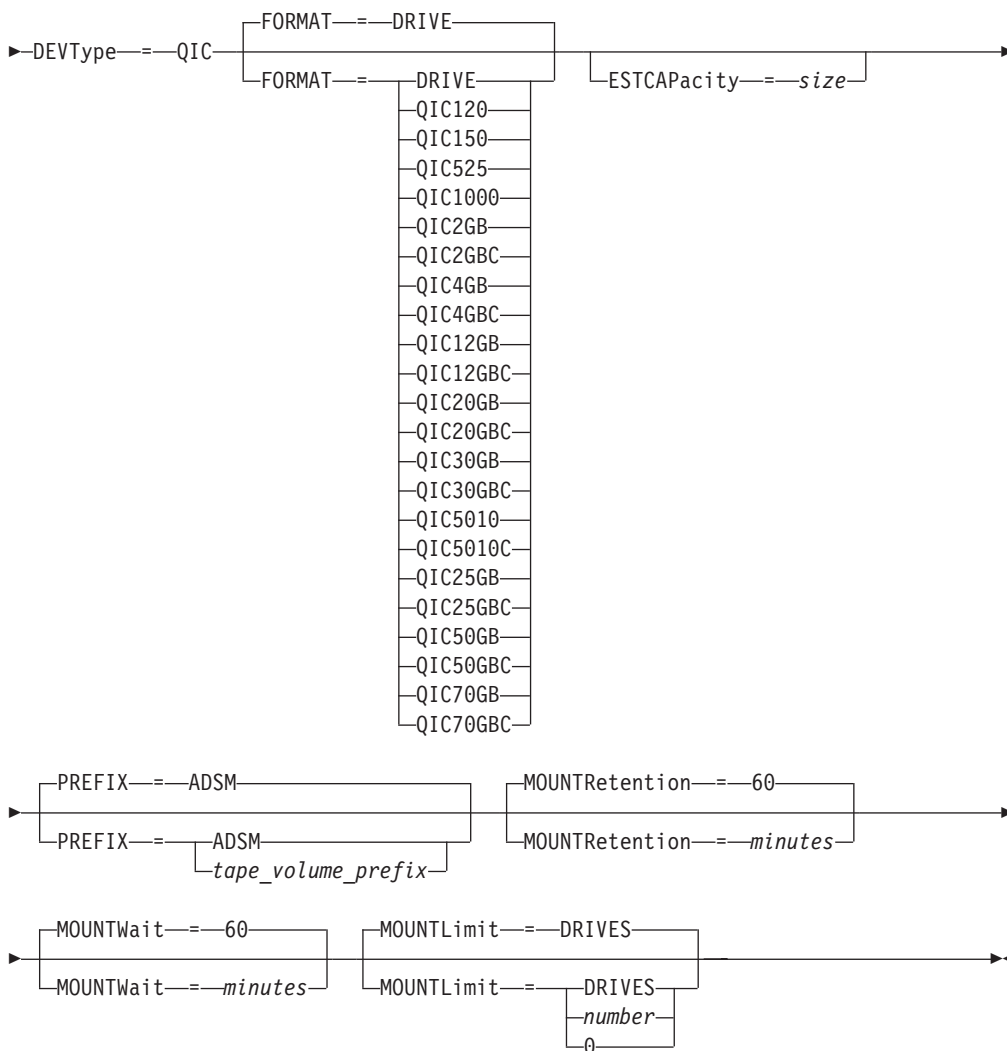
Use the QIC device class when you are using QIC tape devices.

#### Privilege class

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

#### Syntax

```
►►—DEfIne DEVclAss—device_class_name—LIBRary—==—library_name—————►
```



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### LIBRary (Required)

Specifies the name of the defined library object that contains the tape drives

that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

**DEVType=QIC (Required)**

Specifies that the quarter-inch cartridge (QIC) device type is assigned to the device class.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use. Refer to *Administrator's Guide* for information.

The following tables list the recording formats, estimated capacities and recording format options for QIC devices:

*Table 81. Recording format and default estimated capacity for quarter-inch cartridge (QIC) tape*

| Format  | Estimated Capacity | Description  |
|---------|--------------------|--|
| DRIVE   | –                  | The server selects the highest format that can be supported by the sequential access drive on which a volume is mounted.<br><br>Avoid specifying the DRIVE value when a mixture of devices is used within the same library.<br><br>For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| QIC120  | 26 MB – 172 MB     | 120 QIC format, depending on media<br><br>Using DC600XTD, DC6150, DC6320, and DC6525   |
| QIC150  | 31 MB – 207 MB     | 150 QIC format, depending on media<br><br>Using DC600XTD, DC6150, DC6320, and DC6525   |
| QIC525  | 65 MB – 427 MB     | 525 QIC format, depending on media<br><br>Using DC6320 and DC6525  |
| QIC1000 | 169 MB – 1.1 GB    | 1000 QIC format, depending on media<br><br>Using DC9100 and DC9120XL   |
| QIC2GB  | 2 GB               | Uncompressed 2000 QIC format<br><br>Using DC9100 and DC9120XL  |
| QIC2GBC | See note<br>4 GB   | Compressed 2000 QIC format   |
| QIC4GB  | 4 GB               | Uncompressed 4000 QIC format   |
| QIC4GBC | See note<br>8 GB   | Compressed 4000 QIC format   |
| QIC12GB | 12 GB              | Uncompressed 12000 QIC format, using 343-meter tape  |

## DEFINE DEVCLASS — QIC

*Table 81. Recording format and default estimated capacity for quarter-inch cartridge (QIC) tape (continued)*

| Format   | Estimated Capacity        | Description                                       |
|----------|---------------------------|---|
| QIC12GBC | See note<br>24 GB         | Compressed 12000 QIC format, using 343-meter tape |
| QIC5010  | 13 GB – 16 GB             | Uncompressed 5010 QIC format, depending on media  |
| QIC5010C | See note<br>26 GB – 32 GB | Compressed 5010 QIC format, depending on media    |
| QIC20GB  | 20 GB                     | Uncompressed 20000 QIC format                     |
| QIC20GBC | See note<br>40 GB         | Compressed 20000 QIC format                       |
| QIC25GB  | 25 GB                     | Uncompressed 25000 QIC format                     |
| QIC25GBC | See note<br>50 GB         | Compressed 25000 QIC format                       |
| QIC30GB  | 30 GB                     | Uncompressed 30000 QIC format                     |
| QIC30GBC | See note<br>60 GB         | Compressed 30000 QIC format                       |
| QIC50GB  | 50 GB                     | Uncompressed 50GB QIC format                      |
| QIC50GBC | See note<br>100 GB        | Compressed 50GB QIC format                        |
| QIC70GB  | 70 GB                     | Uncompressed 70GB QIC format                      |
| QIC70GBC | See note<br>140 GB        | Compressed 70GB QIC format                        |

**Note:** If 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.

*Table 82. QIC tape recording format options*

| 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:** The server cannot use 3M DC300XLP and 3M DC600A tapes.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for QIC tapes, see Table 81 on page 201.

### **PREFIX**

Specifies the high level qualifier of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Tip:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

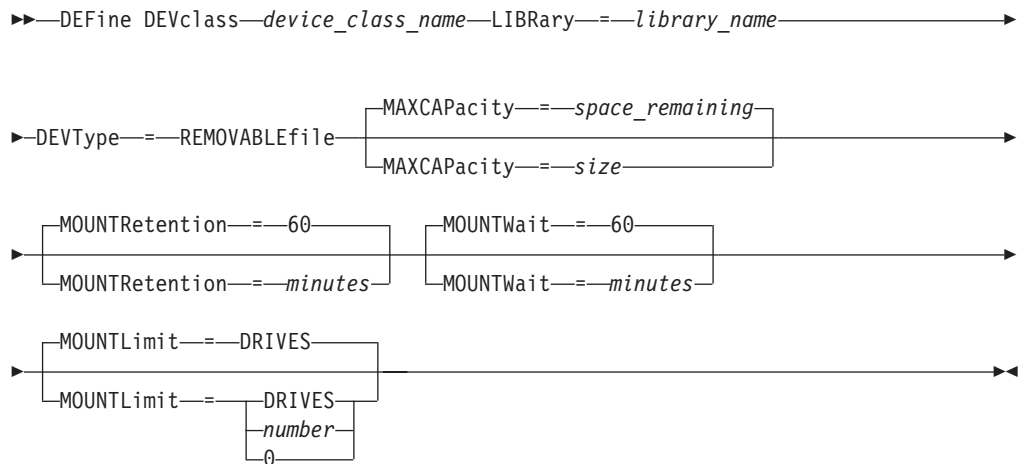
## DEFINE DEVCLASS (Define a REMOVABLEFILE device class)

Use the REMOVABLEFILE device class for removable media devices that are attached as local, removable file systems.

### Privilege class

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

### Syntax



### Parameters

#### device\_class\_name (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY (Required)

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.

#### DEVType=REMOVABLEfile (Required)

Specifies that the REMOVABLEFILE device type is assigned to the device class. REMOVABLEFILE indicates that the volumes for this device class are files on local, removable media.

Volumes in a device class with device type REMOVABLEFILE are sequential access volumes.

Use the device manufacturer's utilities to format (if necessary) and label the media. The label on the media must meet the following restrictions:

- The label can have no more than 11 characters.
- The volume label and the name of the file on the volume must match exactly.
- The MAXCAPACITY parameter value must be specified at less than the capacity of the media.

Refer to the *Administrator's Guide* for more information.

### MAXCAPacity

Specifies the maximum size of any volumes defined to a storage pool categorized by this device class. This parameter is optional.

The MAXCAPACITY parameter must be set at less value than the capacity of the media. For CD media, the maximum capacity should be no greater than 650 MB.

Because the server opens only one file per physical removable medium, specify a capacity that enables one file to make full use of your media capacity.

#### *space\_remaining*

The default maximum capacity is the space remaining on the media after it is first used.

#### *size*

You must specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes).

For example, MAXCAPACITY=5M specifies that the maximum capacity for a volume in this device class is 5 MB. The smallest value allowed is 100 KB (that is, MAXCAPACITY=100K).

### MOUNTRetention

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 a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### MOUNTLimit

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Tip:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### DEFINE DEVCLASS (Define a SERVER device class)

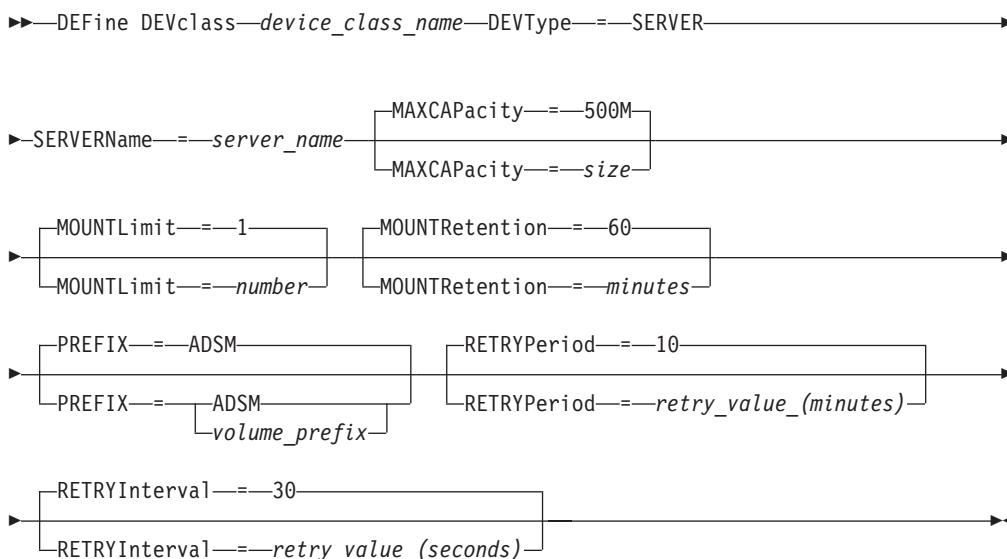
Use the SERVER device class to use storage volumes or files archived in another Tivoli Storage Manager server.

If data retention protection is activated using the SET ARCHIVERETENTIONPROTECTION command, you cannot define a server device class. See the *Administrator's Guide* for more information.

#### Privilege class

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

#### Syntax



#### Parameters

##### device\_class\_name (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### DEVType=SERVER (Required)

Specifies a remote connection that supports virtual volumes.

##### SERVERName (Required)

Specifies the name of the server. The SERVERNAME parameter must match a defined server.

##### MAXCAPacity

Specifies the maximum size for objects created on the target server; the default for this value is 500M. This parameter is optional.

##### 500M

Specifies that the maximum capacity is 500M (500 MB).

##### size

Specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The minimum value allowed is 100 KB (MAXCAPACITY=100K).

### **MOUNTLimit**

Specifies the maximum number of simultaneous sessions between the source server and the target server. Any attempts to access more sessions than indicated by the mount limit cause the requester to wait. This parameter is optional. The default value is 1. You can specify a number from 1 to 4096.

The following are possible values:

- 1 Specifies that only one session between the source server and the target server is allowed.

*number*

Specifies the number of simultaneous sessions between the source server and the target server.

### **MOUNTRetention**

Specifies the number of minutes to retain an idle connection with the target server before closing the connection. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

### **PREFIX**

Specifies the beginning portion of the high-level archive file name on the target server. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a high level archive file name using the default prefix is ADSM.volume1.

### **RETRYPeriod**

Specifies the retry period in minutes. The retry period is the interval during which the server attempts to contact a target server if there is a suspected communications failure. This parameter is optional. The default value is 10 minutes.

### **RETRYInterval**

Specifies the retry interval in seconds. The retry interval is how often retries are done within a given time period. This parameter is optional. The default value is 30 seconds.

### DEFINE DEVCLASS (Define a VOLSAFE device class)

Use the VOLSAFE device type to work with StorageTek VolSafe brand media and drives. This technology uses media that cannot be overwritten. Therefore, do not use these media for short-term backups of client files, the server database, or export tapes.

#### Restrictions:

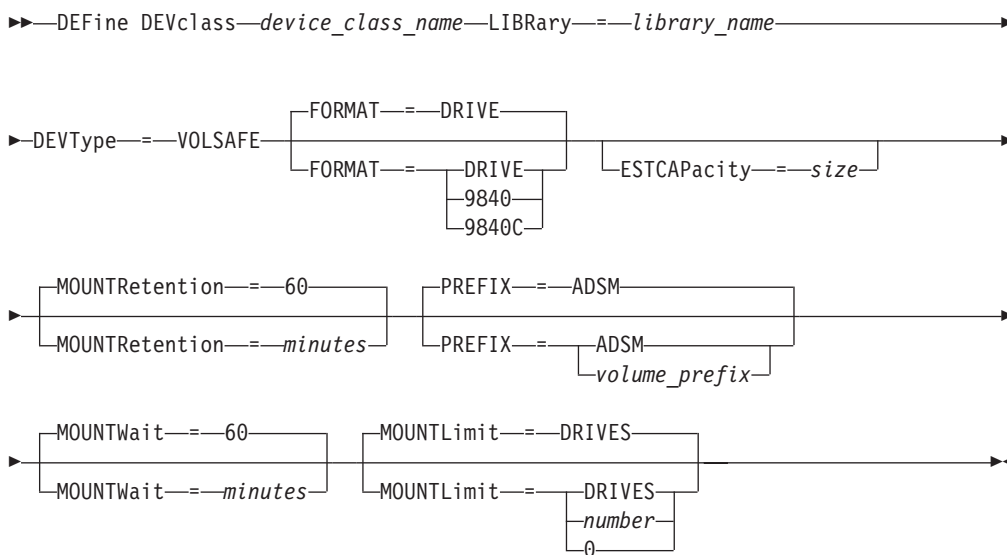
1. NAS-attached libraries are not supported.
2. VolSafe media and read-write media must reside in separate storage pools.
3. Check in cartridges with CHECKLABEL=YES on the CHECKIN LIBVOLUME command.
4. Label cartridges with OVERWRITE=NO on the LABEL LIBVOLUME command. If VolSafe cartridges are labeled more than once, no additional data can be written to them.

See the *Administrator's Guide* for additional information.

#### Privilege class

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

#### Syntax



#### Parameters

##### device\_class\_name (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### LIBRARY (Required)

Specifies the name of the defined library object that contains the VolSafe drives that can be used by this device class. If any drives in a library are VolSafe-enabled, all drives in the library must be VolSafe-enabled. Consult your hardware documentation to enable VolSafe on the 9840 drives.



For information about defining a library object, see “DEFINE LIBRARY (Define a library)” on page 222.

#### DEVType=VOLSAFE (Required)

Specifies that the VOLSAFE device type is assigned to the device class. The label on this type of cartridge can only be overwritten one time, which Tivoli Storage Manager does when it writes the first block of data. Therefore, it is important to limit the use of the LABEL LIBVOLUME command to one time per volume by using the OVERWRITE=NO parameter.

#### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

**Important:** If you specify DRIVE for a device class that has non-compatible sequential access devices, then you must mount volumes on devices that are capable of reading or writing the format established when the volume was first mounted. This can cause delays if the only sequential access device that can access the volume is already in use.

The following table lists the recording formats and estimated capacities for VolSafe devices:

Table 83. Recording formats and default estimated capacities for Volsafe media

| Format | Estimated Capacity | Description  |
|--------|--------------------|--|
| DRIVE  | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 9840   | 20 GB              | Uncompressed (standard) format, using a 20 GB cartridge with 270 meters (885 feet) of tape   |
| 9840C  | See note<br>80 GB  | LZ-1 Enhanced (4:1) compressed format, using a 80 GB cartridge with 270 meters (885 feet) of tape  |

#### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for cartridge tapes, see Table 83.

### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### PREFIX

Specifies the beginning portion of the high-level archive file name on the target server. This parameter is optional. The default value is ADSM. The maximum length of this prefix is 8 characters.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a high level archive file name using the default prefix is ADSM.volume1.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### MOUNTLimit

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Tip:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### DEFINE DOMAIN (Define a new policy domain)

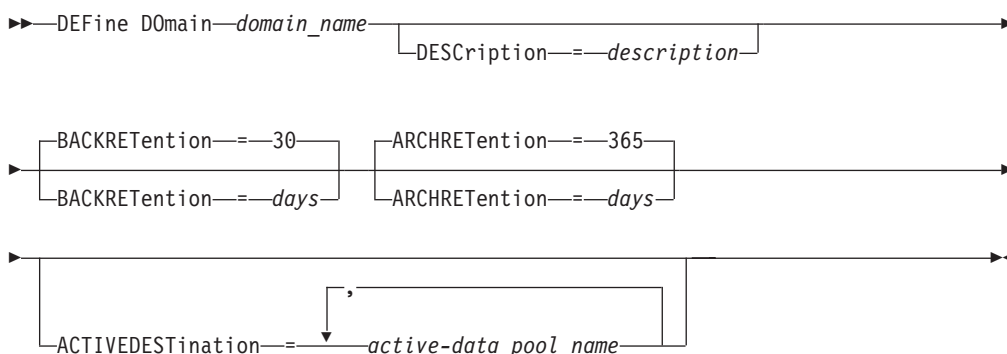
Use this command to define a new policy domain. A policy domain contains policy sets, management classes, and copy groups. A client is assigned to one policy domain. The ACTIVE policy set in the policy domain determines the rules for clients assigned to the domain. The rules control the archive, backup, and space management services provided for the clients.

You must activate a policy set in the domain before clients assigned to the policy domain can back up, archive, or migrate files.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax



#### Parameters

##### *domain\_name* (Required)

Specifies the name of the policy domain to be defined. The maximum length of this name is 30 characters.

##### DESCRiption

Specifies a description of the policy domain. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

##### BACKRETention

Specifies the number of days (from the date the backup versions became inactive) to retain backup versions of files that are no longer on the client file system. This parameter is optional. You can specify an integer from 0 to 9999. The default value is 30. The server uses the backup retention value to manage inactive versions of files when any of the following conditions occur:

- A file is rebound to a new management class, 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

Specifies the number of days (from the date of archive) to retain archive copies. This parameter is optional. You can specify an integer from 0 to 30000. The default value is 365. The server uses the archive retention value to manage archive copies of files when either of the following conditions occur:

- 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.

### ACTIVEDESTination

Specifies the names of active-data pools that store active versions of backup data for nodes assigned to the domain. This parameter is optional. Spaces between the names of the active-data pools are not permitted. You cannot specify more than ten active-data pools for a domain.

Before the Tivoli Storage Manager server writes data to an active-data pool, it verifies that the node owning the data is assigned to a domain that has the active-data pool listed in the **ACTIVEDESTINATION** list. If the server verifies that the node meets this criteria, the data is stored in the active-data pool. If the node does not meet the criteria, then the data is not stored in the active-data pool. If the backup is performed with simultaneous write to an active-data pool, the server performs the verification during backup operations by Tivoli Storage Manager backup-archive clients or by application clients using the Tivoli Storage Manager API. The verification is also performed when active-data is being copied using the **COPY ACTIVEDATA** command.

## Example: Define a policy domain

Define a policy domain with a name of **PROG1** and the description, Programming Group Domain. Specify that archive copies are retained 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 versions are retained for 60 days when management classes or copy groups are deleted and the default management class does not contain a backup copy group.

```
define domain prog1
description="Programming Group Domain"
backretention=60 archretention=90
```

## Related commands

Table 84. Commands related to **DEFINE DOMAIN**

| Command                   | Description   |
|---------------------------|---|
| <b>ACTIVATE POLICYSET</b> | Validates and activates a policy set.                                       |
| <b>COPY DOMAIN</b>        | Creates a copy of a policy domain.  |
| <b>DEFINE POLICYSET</b>   | Defines a policy set within the specified policy domain.                    |
| <b>DELETE DOMAIN</b>      | Deletes a policy domain along with any policy objects in the policy domain. |
| <b>QUERY DOMAIN</b>       | Displays information about policy domains.                                  |
| <b>UPDATE DOMAIN</b>      | Changes the attributes of a policy domain.                                  |

## DEFINE DRIVE (Define a drive to a library)

Use this command to define a drive. Each drive is assigned to a library, and so the library must be defined before you issue this command.

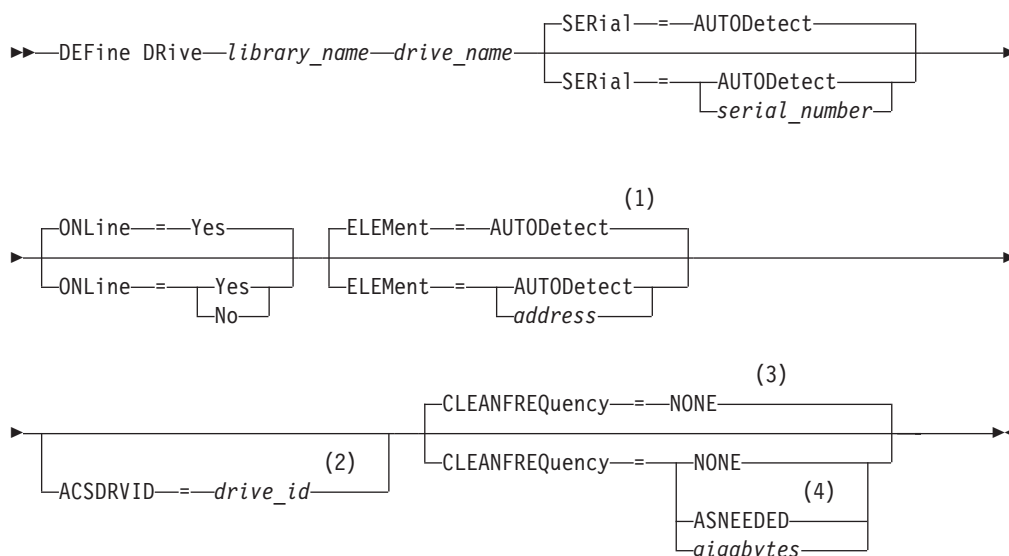
A path must be defined after you issue the DEFINE DRIVE command in order to make the drive usable by Tivoli Storage Manager software. For more information, see “DEFINE PATH (Define a path)” on page 240.

You can define more than one drive for a library by issuing the DEFINE DRIVE command once for each drive. Stand-alone drives always require a manual library. For additional information, see the *Administrator's Guide*.

### Privilege class

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

### Syntax



### Notes:

- 1 The `ELEMENT` parameter is valid only for drives in SCSI libraries. This parameter is not effective when the command is issued from a library client server (that is, when the library type is `SHARED`).
- 2 `ACSDRVID` is required for drives in ACSLS libraries. This parameter is not valid for non-ACSLs libraries.
- 3 The `CLEANFREQUENCY` parameter is valid only for drives in SCSI libraries.
- 4 The `CLEANFREQUENCY=ASNEEDED` parameter value does not work for all tape drives. See the parameter description for more information.

### Parameters

#### *library\_name* (Required)

Specifies the name of the library to which the drive is assigned. This parameter is required for all drives, including stand-alone drives. The specified library must have been previously defined by using the `DEFINE LIBRARY` command.

*drive\_name* **(Required)**

Specifies the name that is assigned to the drive. The maximum length of this name is 30 characters.

**SERial**

Specifies the serial number for the drive being defined. This parameter is optional. The default is AUTODETECT.

If SERIAL=AUTODETECT, then the serial number reported by the drive when you define the path will be used as the serial number.

If SERIAL=*serial\_number*, then the serial number entered will be used to verify that the path to the drive is correct when you define the path.

**Note:** Depending on the capabilities of the device, SERIAL=AUTODETECT may not be supported. In this case, the serial number will be reported as blank.

**ONLine**

Specifies whether the drive is available for use. This parameter is optional. The default is YES.

**Yes**

Specifies that the drive is available for use.

**No**

Specifies that the drive is not available for use.

**ELEMent**

Specifies the element address of the drive within a SCSI library. The server uses the element address to connect the physical location of the drive to the SCSI address of the drive. The default is AUTODETECT.

If ELEMENT=AUTODETECT, then the element number will automatically be detected by Tivoli Storage Manager when the path to the drive is defined.

To find the element address for your library configuration, consult the manufacturer's information.

Additional information on element addresses can be found at the IBM Tivoli Storage Manager Web site: <http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html>. Follow the links to device information. Select the appropriate product and platform, then scroll to the required device. Click on a specific device to see element addresses.

**Restriction:**

- The ELEMENT parameter is valid only for drives in SCSI libraries.
- This parameter is not effective when the command is issued from a library client server (that is, when the library type is SHARED).
- Depending on the capabilities of the library, ELEMENT=AUTODETECT may not be supported. In this case you will have to supply the element address.

**ACSDRVID**

Specifies the ID of the drive that is being accessed in an ACSLS library. The drive ID is a set of numbers that indicates the physical location of a drive within an ACSLS library. This drive ID must be specified as *a,l,p,d*, where *a* is the ACSID, *l* is the LSM (library storage module), *p* is the panel number, and *d* is the drive ID. The server needs the drive ID to connect the physical location of the drive to the drive's SCSI address. See the StorageTek documentation for details.

### CLEANFREQuency

Specifies how often the server activates drive cleaning. This parameter is optional. The default is NONE. For the most complete automation of cleaning for an automated library, you must have a cleaner cartridge checked into the library's volume inventory.

For details about using this parameter for automated and manual libraries, see the *Administrator's Guide*. This parameter is not valid for externally managed libraries, such as 3494 libraries or StorageTek libraries managed under ACSLS.

**Important:** There are special considerations if you plan to use server-activated drive cleaning with a SCSI library that provides automatic drive cleaning support in its device hardware. See the *Administrator's Guide* for details.

### NONE

Specifies that the server does not track cleaning for this drive. This value can be used for libraries that have their own automatic cleaning.

### ASNEEDED

Specifies that the server loads the drive with a checked-in cleaner cartridge only when a drive reports to the device driver that it needs cleaning.

The CLEANFREQUENCY=ASNEEDED parameter value does not work for all tape drives. To determine whether a drive supports this function, see the following Web site:

[www.ibm.com/software/sysmgmt/products/support/IBM\\_TSM\\_Supported\\_Devices\\_for\\_AIXHPSUNWIN.html](http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html)

At the Web site, click the drive to view detailed information. If ASNEEDED is not supported, you can use the *gigabytes* value for automatic cleaning.

**Restriction:** Tivoli Storage Manager does not control the drives connected to the NAS file server. If a drive is attached only to a NAS file server (no connection to a storage agent or server), do not specify ASNEEDED for the cleaning frequency.

### *gigabytes*

Specifies, in gigabytes, how much data is processed on the drive before the server loads the drive with a cleaner cartridge. The server resets the gigabytes-processed counter each time it loads a cleaner cartridge in the drive.

Consult the drive manufacturer's information for cleaning recommendations. If the information gives recommendations for cleaning frequency in terms of hours of use, convert to a gigabytes value by doing the following:

1. Use the bytes-per-second rating for the drive to determine a gigabytes-per-hour value.
2. Multiply the gigabytes-per-hour value by the recommended hours of use between cleanings.
3. Use the result as the cleaning frequency value.

Using the cleaning frequency recommended by IBM for IBM drives will not overclean the drives.

For IBM 3590 and IBM 3570, specify a gigabyte value for the cleaning frequency to ensure that the drives receive adequate cleaning.



### Example: Define a drive to library

Define a drive in a manual library with a library name of LIB01 and a drive name of DRIVE01.

```
define drive lib01 drive01
define path server01 drive01 srctype=server desttype=drive
library=lib01 device=/dev/rmt0
```

### Example: Define a drive in an ACSLS library

Define a drive in an ACSLS library with a library name of ACSLIB and a drive name of ACSDRV1.

```
define drive acslib acsdrv1 acsdrv1 acsdrv1=1,2,3,4
define path server01 acsdrv1 srctype=server desttype=drive
library=acslib device=/dev/rmt0
```

### Example: Define a drive in an automated library

Define a drive in an automated library with a library name of AUTO8MMLIB and a drive name of DRIVE01.

```
define drive auto8mmlib drive01 element=82
define path server01 drive01 srctype=server desttype=drive
library=auto8mmlib device=/dev/rmt0
```

## Related commands

Table 85. Commands related to DEFINE DRIVE

| Command        | Description   |
|----------------|---|
| DEFINE LIBRARY | Defines an automated or manual library.                             |
| DEFINE PATH    | Defines a path from a source to a destination.                      |
| DELETE DRIVE   | Deletes a drive from a library.                                     |
| DELETE LIBRARY | Deletes a library.  |
| QUERY DRIVE    | Displays information about drives.                                  |
| QUERY LIBRARY  | Displays information about one or more libraries.                   |
| QUERY PATH     | Displays information about the path from a source to a destination. |
| UPDATE DRIVE   | Changes the attributes of a drive.                                  |
| UPDATE PATH    | Changes the attributes associated with a path.                      |

### DEFINE EVENTSERVER (Define a server as the event server)

Use this command to identify a server as the event server.

If you define an event server, one Tivoli Storage Manager server can send events to another Tivoli Storage Manager server that will log those events. See the *Administrator's Guide* for information about setting up logging events to an event server.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

►►—DEFINE EVENTSERVER—*server\_name*—◄◄

#### Parameters

##### *server\_name* (Required)

Specifies the name of the event server. The server you specify must have already been defined with the DEFINE SERVER command.

#### Example: Designate the event server

Designate ASTRO to be the event server.

```
define eventserver astro
```

#### Related commands

Table 86. Commands related to DEFINE EVENTSERVER

| Command            | Description   |
|--------------------|---|
| DEFINE SERVER      | Defines a server for server-to-server communications. |
| DELETE EVENTSERVER | Deletes reference to the event server.                |
| DISABLE EVENTS     | Disables specific events for receivers.               |
| ENABLE EVENTS      | Enables specific events for receivers.                |
| PING SERVER        | Test the connections between servers.                 |
| QUERY EVENTSERVER  | Displays the name of the event server.                |
| QUERY SERVER       | Displays information about servers.                   |

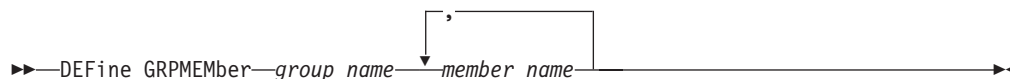
## DEFINE GRPMEMBER (Add a server to a server group)

Use this command to add a server as a member of a server group. You can also add one server group to another server group. A server group lets you route commands to multiple servers by specifying only the server group name.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *group\_name* (Required)

Specifies the name of the server group to which the member will be added.

#### *member\_name* (Required)

Specifies the names of the servers or groups to be added to the group. To specify multiple servers and groups, separate the names with commas and no intervening spaces. The servers or server groups must already be defined to the server.

### Example: Define a server to a server group

Define the server SANJOSE to server group CALIFORNIA.

```
define grpmember california sanjose
```

### Example: Define a server and a server group to a server group

Define the server TUCSON and the server group CALIFORNIA to server group WEST\_COMPLEX.

```
define grpmember west_complex tucson,california
```

### Related commands

Table 87. Commands related to DEFINE GRPMEMBER

| Command            | Description   |
|--------------------|---|
| DEFINE SERVER      | Defines a server for server-to-server communications. |
| DEFINE SERVERGROUP | Defines a new server group.                           |
| DELETE GRPMEMBER   | Deletes a server from a server group.                 |
| DELETE SERVERGROUP | Deletes a server group.                               |
| MOVE GRPMEMBER     | Moves a server group member.                          |
| QUERY SERVER       | Displays information about servers.                   |
| RENAME SERVERGROUP | Renames a server group.                               |
| UPDATE SERVERGROUP | Updates a server group.                               |

## DEFINE LIBRARY

### DEFINE LIBRARY (Define a library)

Use this command to define a library. 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.

A library can only be accessed by one source. This can be either an IBM Tivoli Storage Manager server or a data mover. However, the drives in a library can be accessed by multiple sources.

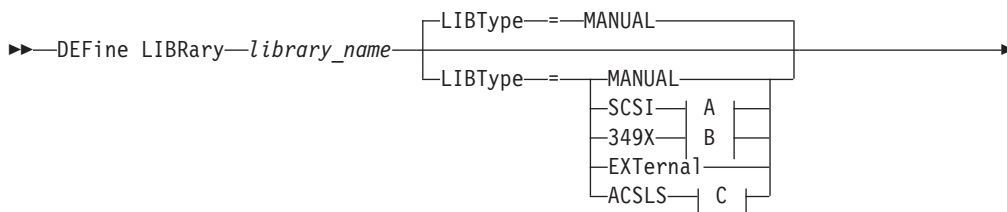
This section includes syntax diagrams for a number of different library configurations, on a LAN or storage area network (SAN):

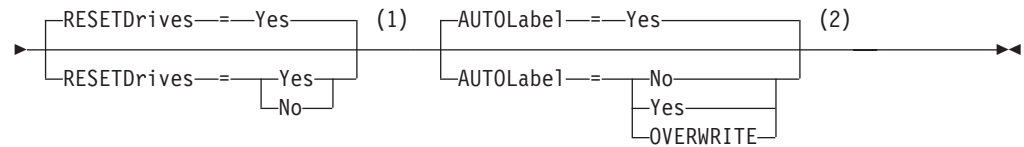
| Configuration task  | Syntax diagram   |
|---|--|
| LAN - Define a library (MANUAL, SCSI, 349X, EXTERNAL, ACSLS).   | "Syntax for a library in a LAN, not used for NDMP operations"  |
| SAN - Define a library (SCSI, 349X, FILE, ACSLS) to a library manager server.   | "Syntax for a library in a SAN, not used for NDMP operations (define a library to a library manager)" on page 223 (SAN - Define a library to a library manager)  |
| SAN - Define a library (SHARED) to a library client server.   | "Syntax for a library in a SAN, not used for NDMP operations (define a library to a library client)" on page 224 (SAN - Define a library to a library client)  |
| SAN or LAN - Define a SCSI library that will be accessed by a NAS data mover and directly controlled by Tivoli Storage Manager. | "Syntax for a library in a LAN or a SAN, used for NDMP operations (define a library controlled directly by Tivoli Storage Manager)" on page 224 (Define a library controlled directly by Tivoli Storage Manager) |
| SAN or LAN - Define a SCSI library to be accessed by NAS data mover and controlled through a NAS file server.                   | "Syntax for a library in a LAN or a SAN, used for NDMP operations (define a library controlled directly by Tivoli Storage Manager)" on page 224 (Define a library controlled through a NAS file server)          |
| SAN - Define an EXTERNAL library.   | "Syntax for an EXTERNAL library shared with storage agents" on page 225  |

### Privilege class

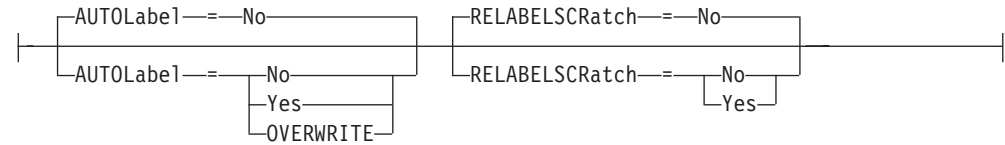
To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax for a library in a LAN, not used for NDMP operations

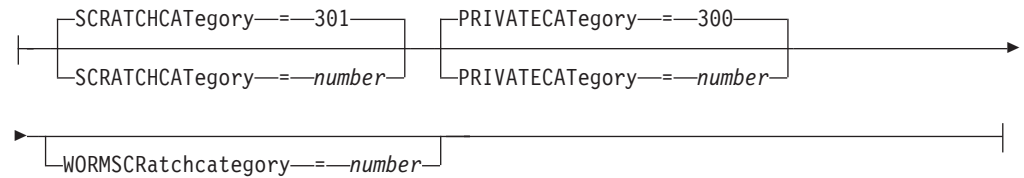




### A (SCSI):



### B (349X):



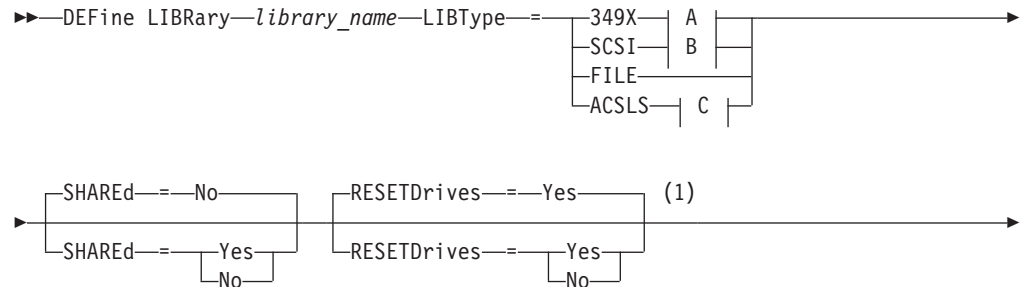
### C (ACSL):



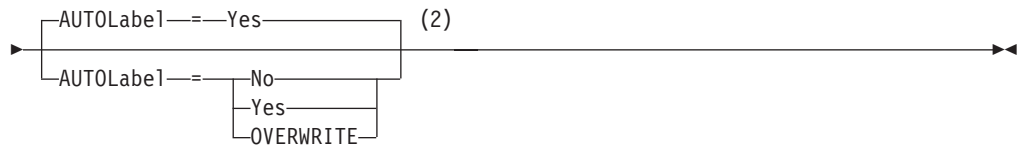
### Notes:

- 1 RESETDRIVES defaults to YES only when a library is defined with SHARED=YES; otherwise, the parameter defaults to NO.
- 2 AUTOLABEL defaults to YES for all non-SCSI libraries and to NO for SCSI libraries.

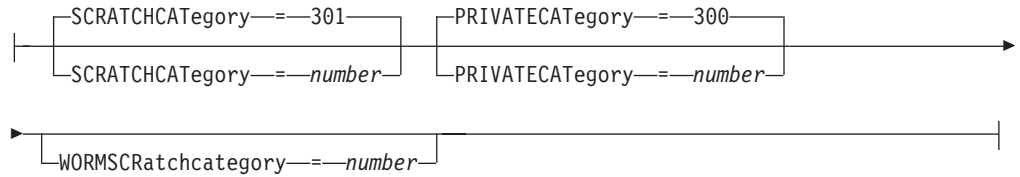
### Syntax for a library in a SAN, not used for NDMP operations (define a library to a library manager)



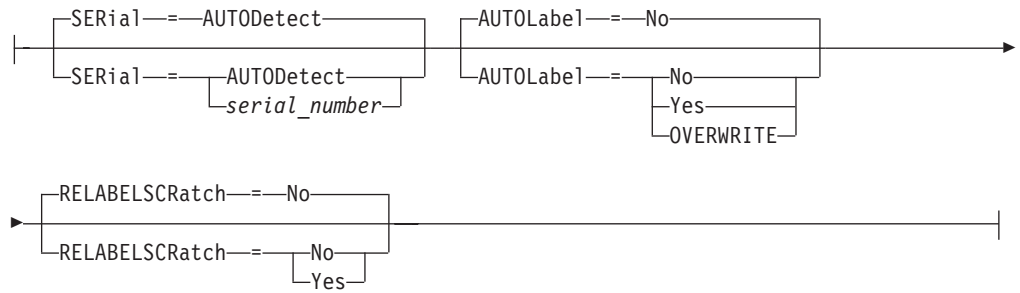
## DEFINE LIBRARY



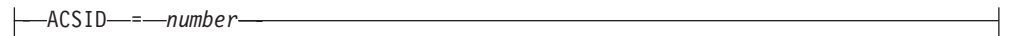
**A (349X):**



**B (SCSI):**



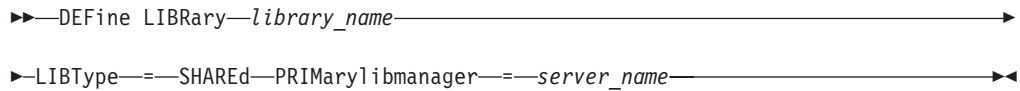
**C (ACSLs):**



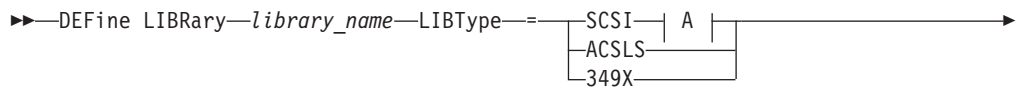
**Notes:**

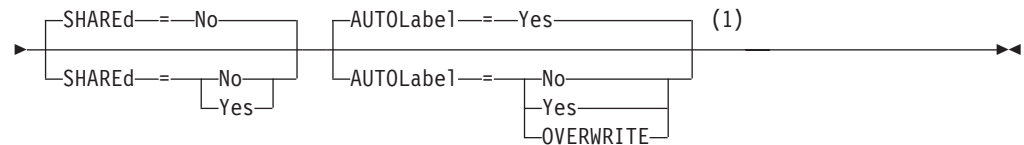
- 1 RESETDRIVES defaults to YES only when a library is defined with SHARED=YES; otherwise, the parameter defaults to No.
- 2 AUTOLABEL defaults to YES for all non-SCSI libraries and to NO for SCSI libraries.

### Syntax for a library in a SAN, not used for NDMP operations (define a library to a library client)

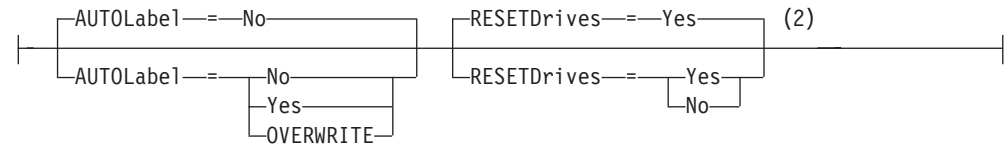


**Syntax for a library in a LAN or a SAN, used for NDMP operations (define a library controlled directly by Tivoli Storage Manager)**





### A (SCSI):



### Notes:

- 1 AUTOLABEL defaults to YES for all non-SCSI libraries and to NO for SCSI libraries.
- 2 RESETDRIVES defaults to YES only when a library is defined with SHARED=YES; otherwise, the parameter defaults to No.

### Syntax for a library in a LAN or a SAN, used for NDMP operations (define a library controlled through a NAS file server)

```

>>> DEFINE LIBRARY library_name LIBType=SCSI SHARED=No >>>
  
```

### Syntax for an EXTERNAL library shared with storage agents

```

>>> DEFINE LIBRARY library_name LIBType=EXTERNAL >>>
  
```

### Parameters

#### library\_name (Required)

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### LIBType

Specifies the type of library that is being defined. The default is MANUAL. Possible values are:

##### MANUAL

Specifies that the library is not automated. When volumes must be mounted on drives in this type of library, messages are sent to operators. This type of library is used with stand-alone drives.

##### FILE

Specifies that a pseudo-library is created for sequential file volumes. When you issue the DEFINE DEVCLASS command with DEVTYPE=FILE and SHARED=YES parameters, this occurs automatically. FILE libraries are necessary only when sharing sequential file volumes between the server and one or more storage agents. The use of FILE libraries requires library sharing.

## DEFINE LIBRARY

### SCSI

Specifies that the library has a SCSI-controlled media changer device. To mount volumes on drives in this type of library, Tivoli Storage Manager uses the media changer device.

### 349X

Specifies that the library is an IBM 3494 or 3495 Tape Library Dataserver. For more information about specifying category numbers for scratch, private, and WORM volumes, see the *Administrator's Guide*.

**Restriction:** IBM 3494 libraries support only one unique device type at a time.

If you currently have an IBM 3494 library with both 3490 and 3590 drives, you will need to follow the upgrade procedure explained in “Configuring Storage Devices” in the *Administrator's Guide*. This procedure will separate the library into two distinct library objects.

### SHARED

Specifies that the library is shared with another Tivoli Storage Manager server over a storage area network (SAN) or a dual SCSI connection to library drives. This library type is not valid for optical devices.

**Important:** Specify this library type when defining the library on the library client.

### EXTERNAL

Specifies that the library is managed by an external media management system. This library type does not support drive definitions with the DEFINE DRIVE command. Rather, the external media management system identifies the appropriate drive for media access operations.

In an IBM Tivoli Storage Manager for Storage Area Networks environment, this parameter specifies that StorageTek Automated Cartridge System Library Software (ACSL) or Library Station software controls the library. Software, such as Gresham EDT-Distributable, allows multiple servers to share the library. The drives in this library are not defined to Tivoli Storage Manager. ACSL identifies the drive for media operations.

### ACSL

Specifies that the library is a StorageTek library that is controlled by StorageTek Automated Cartridge System Library Software (ACSL).

### SERIAL

Specifies the serial number for the library being defined. This parameter is optional. The default is AUTODETECT.

If SERIAL=AUTODETECT, then when you define the path to the library, the serial number reported by the library will be used as the serial number.

If SERIAL=*serial\_number*, then the number you have entered will be compared to the number detected by Tivoli Storage Manager.

**Attention:** Depending on the capabilities of the device, SERIAL=AUTODETECT may not be supported. In this case, the serial number will be reported as blank.

### AUTOLabel

Specifies whether the server attempts to automatically label tape volumes. This



parameter is optional. The default for 349X, ACSLS, EXTERNAL, and MANUAL libraries is YES. The default for SCSI libraries is NO.

To use this option, you need to check in the tapes with CHECKLABEL=BARCODE on the CHECKIN LIBVOLUME command.

### No

Specifies that the server does not attempt to label any volumes.

### Yes

Specifies that the server only labels unlabeled volumes.

### OVERWRITE

Specifies that the server attempts to overwrite an existing label. The server overwrites existing labels *only* if both the existing label and the bar code label are not already defined in any server storage pool or volume history list.

### RELABELSCRatch

Specifies whether the server relabels volumes that have been deleted and returned to scratch. When this parameter is set to YES, a LABEL LIBVOLUME operation is started and the existing volume label is overwritten. This parameter is optional and intended for use with a Virtual Tape Library (VTL).

**Note:** If you have both virtual and real volumes in your VTL, both types will be relabeled when this parameter is enabled. If the VTL includes real volumes, specifying this option might impact performance.

### No

Specifies that the server does not relabel volumes that are deleted and returned to scratch.

### Yes

Specifies that the server relabels volumes that are deleted and returned to scratch.

### ACSID

Specifies the number of this StorageTek library assigned by the ACSSA (Automatic Cartridge System System Administrator). This can be a number from 0 to 126. Issue QUERY ACS on your system to get the number for your library ID. This parameter is required and valid only when LIBTYPE=ACSLs.

See your StorageTek documentation for more information.

### PRIMarylibmanager

Specifies the name of the Tivoli Storage Manager server that is responsible for controlling access to library resources. You must define this server with the DEFINE SERVER command before you can use it as a library manager. This parameter is required and valid only if LIBTYPE=SHARED (that is, when you define a library in a SAN to a library client server).

### PRIVATECAtegory

Specifies the category number for private volumes that must be mounted by name. This parameter is optional. The default value is 300 (this value becomes X'12C' on the IBM 3494 because it uses hexadecimal values). You can specify a number from 1 to 65279. This number must be unique. It cannot be shared with other applications or defined libraries, and it must be different than the other category numbers in this library. This parameter is valid only when LIBTYPE=349X.

### SCRATCHCAtegory

Specifies the category number to be used for scratch volumes in the library.

## DEFINE LIBRARY

This parameter is optional. The default value is 301 (becomes X'12D' on the IBM 3494 since it uses hexadecimal values). You can specify a number from 1 to 65279. This number must be unique. It cannot be shared with other applications or defined libraries, and it must be different than the other category numbers in this library. This parameter is valid only when LIBTYPE=349X.

### **WORMSCRatchcategory**

Specifies the category number to be used for WORM scratch volumes in the library. This parameter is required if you use WORM volumes. You can specify a number from 1 to 65279. This number must be unique. It cannot be shared with other applications or defined libraries, and it must be different than the other category numbers in this library. This parameter is only valid when LIBTYPE=349X and 3592 WORM volumes are used.

**Restriction:** If a 349X library does not have the WORMSCRATCHCATEGORY defined and the **WORM** parameter is set to YES for the device class, the mount operation will fail with an error message.

### **SHARED**

Specifies whether this library is shared with other Tivoli Storage Manager servers in a storage area network (SAN). This parameter is required when you define a library to the library manager. This parameter is valid only when you define a SCSI, 349X, or ACSLS library.

#### **YES**

Specifies that this library can be shared with other servers. When you specify YES, the library manager server mounts volumes as requested by other servers and tracks drive and volume allocation to other servers.

#### **NO**

Specifies that this library cannot be shared with other servers. SHARED=NO is required if the library is controlled by passing commands through a NAS file server.

### **RESETDrives**

Specifies whether the server performs a target reset when the server is restarted or when a library client or storage agent re-connection is established. This parameter only applies to SCSI, 3494, MANUAL, and ACSLS type libraries.

#### **Yes**

Specifies that the target reset is to be performed. YES is the default for SCSI, 3494, MANUAL, and ACSLS libraries defined with SHARED=YES.

#### **No**

Specifies that the target reset is not performed. NO is the default for SCSI, 3494, MANUAL, and ACSLS libraries defined with SHARED=NO.

## **Example: Define a manual library**

Define a library named MANUALMOUNT with the library type of MANUAL.

```
define library manualmount libtype=manual
```

## **Example: Define a SCSI library**

Define a library named SCسيلIB with a library type of SCSI.

```
define library scsilib libtype=scsi
```

The library requires a path. The device name for the library is:

```
/dev/lb0
```

Define the path:

```
define path server1 scsilib srctype=server desttype=library
device=/dev/lb0
```

### Example: Define a SCSI library on a SAN

In a storage area network, two Tivoli Storage Manager servers, CASTOR and POLLUX, must share a SCSI library. Define a SCSI library named LTOLIB, with CASTOR as the library manager server.

1. On the library manager server, CASTOR, define the library as a shared library.

```
define library ltolib libtype=scsi shared=yes
```

2. On the library manager server, CASTOR, define the path. The device name for the library is:

```
/dev/smc0
```

Issue the command to define the path:

```
define path castor ltolib srctype=server desttype=library
device=/dev/smc0
```

3. On the library manager client, POLLUX, define the library as a shared library. The server CASTOR is identified as the library manager.

```
define library ltolib libtype=shared primarylibmanager=castor
```

### Example: Define a shared ACSLS library

Define a library named ACSLIB with a library type of ACSLS and an ACSID of 1.

```
define library acslib libtype=acsls acsid=1 shared=yes
```

### Example: Define a library to be used for NDMP operations

A SCSI library named TSMLIB is connected to a Tivoli Storage Manager server. Define that library so that it will be directly controlled by Tivoli Storage Manager and used for NDMP operations.

1. Define the library:

```
define library tsmlib libtype=scsi
```

2. Define the path. The device name for the library is:

```
/dev/lb0
```

Issue the command to define the path:

```
define path server1 tsmlib srctype=server desttype=library
device=/dev/lb0
```

### Example: Define an ACSLS library to be used for NDMP operations

An ACSLS library named STKLIB is connected to a Tivoli Storage Manager server. Define that library so that it will be directly controlled by Tivoli Storage Manager and used for NDMP operations.

```
define library stklib libtype=acsls acsid=4
```

## DEFINE LIBRARY

### Example: Define a SCSI library to be controlled by a NAS data mover and used for NDMP operations

A SCSI library named NASLIB is connected directly to a NAS file server. Define that library so that it will be controlled by Tivoli Storage Manager through the NAS data mover (named NASDM) and used for NDMP operations.

1. Define the library:

```
define library naslib libtype=scsi
```

2. Define the path:

```
define path nasdm naslib srctype=datamover desttype=library  
device=mc1
```

### Example: Define an external library for a SAN configuration

For an IBM Tivoli Storage Manager for Storage Area Networks configuration, define a library named EXTLIB with the library type of EXTERNAL. If using Gresham Enterprise DistribuTAPE, the external library manager executable file is located in the following directory:

```
/usr/lpp/dtelm/bin/elm
```

1. Define the library:

```
define library extlib libtype=external
```

2. Define the path:

```
define path server1 extlib srctype=server desttype=library  
externalmanager="/usr/lpp/dtelm/bin/elm"
```

## Related commands

Table 88. Commands related to DEFINE LIBRARY

| Command        | Description   |
|----------------|---|
| AUDIT LIBRARY  | Ensures that an automated library is in a consistent state.         |
| DEFINE DRIVE   | Assigns a drive to a library.                                       |
| DEFINE PATH    | Defines a path from a source to a destination.                      |
| DEFINE SERVER  | Defines a server for server-to-server communications.               |
| DELETE DRIVE   | Deletes a drive from a library.                                     |
| DELETE PATH    | Deletes a path from a source to a destination.                      |
| DELETE LIBRARY | Deletes a library.  |
| QUERY DRIVE    | Displays information about drives.                                  |
| QUERY LIBRARY  | Displays information about one or more libraries.                   |
| QUERY PATH     | Displays information about the path from a source to a destination. |
| UPDATE DRIVE   | Changes the attributes of a drive.                                  |
| UPDATE LIBRARY | Changes the attributes of a library.                                |
| UPDATE PATH    | Changes the attributes associated with a path.                      |

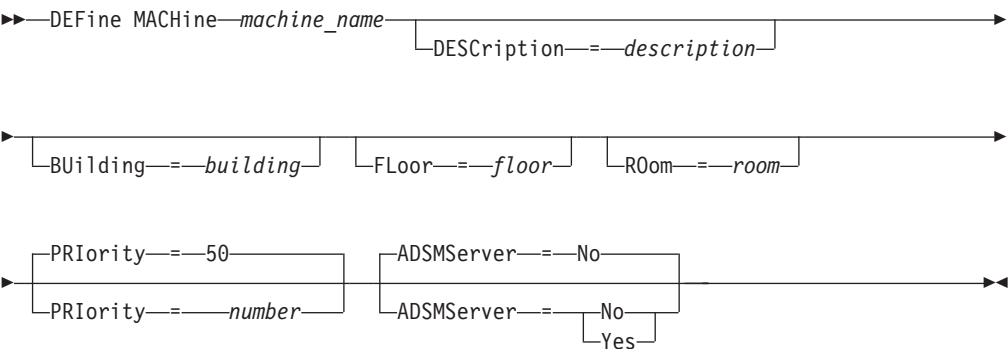
**DEFINE MACHINE (Define machine information for disaster recovery)**

Use this command to save disaster recovery information for a server or client node machine. This information will be included in the plan file to help you recover your machines.

**Privilege class**

To issue this command, you must have system privilege.

**Syntax**



**Parameters**

- machine\_name (Required)**  
Specifies the machine name. The name can be up to 64 characters.
- DESCRIPTION**  
Specifies a machine description. This parameter is optional. The text can be up to 255 characters. Enclose the text in quotation marks if it contains any blank characters.
- BUILDING**  
Specifies the building that this machine is in. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters.
- FLOOR**  
Specifies the floor that this machine is on. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters.
- ROOM**  
Specifies the room that this machine is in. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters.
- PRIORITY**  
Specifies the restore priority for the machine an integer from 1 to 99. The highest priority is 1. This parameter is optional. The default is 50.
- ADSMSEVER**  
Specifies whether the machine is a Tivoli Storage Manager server. Only one machine can be defined as a Tivoli Storage Manager server. This parameter is optional. The default is NO. Possible values are:

## DEFINE MACHINE

### No

This machine is not a Tivoli Storage Manager server.

### Yes

This machine is a Tivoli Storage Manager server.

## Example: Define a machine's disaster recovery information

Define a machine named DISTRICT5, and specify a location, a floor, and a room name. This machine contains critical data and has the highest priority.

```
define machine district5 building=101 floor=27  
room=datafacilities priority=1
```

## Related commands

*Table 89. Commands related to DEFINE MACHINE*

| Command                      | Description  |
|------------------------------|--|
| DEFINE MACHNODEASSOCIATION   | Associates an IBM Tivoli Storage Manager node with a machine.  |
| DEFINE RECMEDMACHASSOCIATION | Associates recovery media with a machine.  |
| DELETE MACHINE               | Deletes a machine.   |
| INSERT MACHINE               | Inserts machine characteristics or recovery instructions into the IBM Tivoli Storage Manager database. |
| QUERY MACHINE                | Displays information about machines.   |
| UPDATE MACHINE               | Changes the information for a machine.   |

## DEFINE MACHNODEASSOCIATION (Associate a node with a machine)

Use this command to associate client nodes with a machine. During disaster recovery, you can use this information to identify the client nodes that resided on destroyed machines.

The machine must be defined and the nodes registered to Tivoli Storage Manager

To retrieve the information, issue the QUERY MACHINE command. This information will be included in the plan file to help you recover the client machines.

A node remains associated with a machine unless the node, the machine, or the association itself is deleted.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>> DEFINE MACHNODEASSOCIATION machine_name node_name
  
```

### Parameters

#### *machine\_name* (Required)

Specifies the machine name.

#### *node\_name* (Required)

Specifies the node names. A node can only be associated with one machine. To specify multiple nodes, separate the names with commas and no intervening spaces. You can use wildcard characters to specify a name.

### Example: Associate a node with a machine

Associate the node named ACCOUNTSPAYABLE with the machine named DISTRICT5.

```
define machnodeassociation district5 accountspayable
```

### Related commands

Table 90. Commands related to DEFINE MACHNODEASSOCIATION

| Command                    | Description  |
|----------------------------|--|
| DEFINE MACHINE             | Defines a machine for DRM.                                     |
| DELETE MACHINE             | Deletes a machine.   |
| DELETE MACHNODEASSOCIATION | Deletes association between a machine and node.                |
| QUERY MACHINE              | Displays information about machines.                           |
| REGISTER NODE              | Defines a client to the server and sets options for that user. |

## DEFINE MACHNODEASSOCIATION

*Table 90. Commands related to DEFINE MACHNODEASSOCIATION (continued)*

| Command     | Description  |
|-------------|--|
| REMOVE NODE | Removes a client from the list of registered nodes for a specific policy domain. |



## DEFINE MGMTCLASS (Define a management class)

Use this command to define a new management class in a policy set. To allow clients to use the new management class, you must activate the policy set that contains the new class.

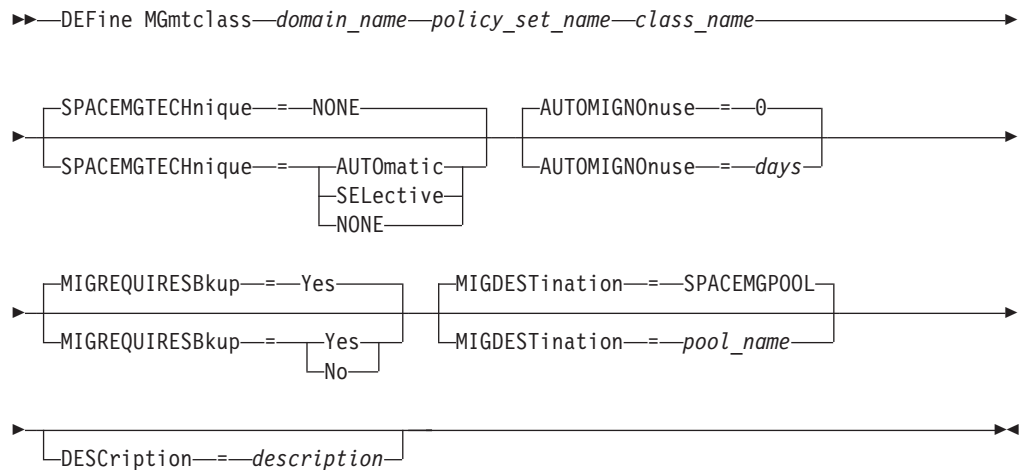
You can define one or more management classes for each policy set in a policy domain. A management class can contain a backup copy group, an archive copy group, or both. The user of a client node can select any management class in the active policy set or use the default management class.

**Attention:** The DEFINE MGMTCLASS command fails if a copy storage pool is specified as the destination for files that were migrated by a Tivoli Storage Manager for Space Management client.

### 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



### Parameters

#### *domain\_name* (Required)

Specifies the policy domain to which the management class belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set to which the management class belongs. You cannot define a management class to the ACTIVE policy set.

#### *class\_name* (Required)

Specifies the name of the new management class. The maximum length of this name is 30 characters. You cannot use either *default* or *grace\_period* as a class name.

#### SPACEMGTECHnique

Specifies whether a file using this management class is eligible for migration. This parameter is optional. The default is NONE. This parameter is effective

## DEFINE MGMTCLASS

only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients. Possible values are:

### **AUTOmatic**

Specifies that the file is eligible for both automatic migration and selective migration.

### **SElective**

Specifies that the file is eligible for selective migration only.

### **NONE**

Specifies that the file is not eligible for migration.

### **AUTOMIGNOnuse**

Specifies the number of days that must elapse since a file was last accessed before it is eligible for automatic migration. This parameter is optional. The default value is 0. If SPACEMGTECHNIQUE is not AUTOMATIC, the server ignores this attribute. You can specify an integer from 0 to 9999.

This parameter is effective only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients.

### **MIGREQUIRESBkup**

Specifies whether a backup version of a file must exist before a file can be migrated. This parameter is optional. The default is YES. This parameter is effective only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients. Possible values are:

#### **Yes**

Specifies that a backup version must exist.

#### **No**

Specifies that a backup version is optional.

### **MIGDESTination**

Specifies the primary storage pool where the server initially stores files migrated by Tivoli Storage Manager for Space Management clients. This parameter is effective only for Tivoli Storage Manager for Space Management clients, and is not effective for backup-archive clients or application clients. The default is SPACEMGPOOL.

The command fails if you specify a copy storage pool as the destination.

### **DESCription**

Specifies a description of the management class. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

## **Example: Define a management class for a specific policy set and policy domain**

Define a management class called MCLASS1 for policy set SUMMER in the PROG1 policy domain. For Tivoli Storage Manager for Space Management clients, allow both automatic and selective migration, and store migrated files in the SMPPOOL storage pool. Add the description, "Technical Support Mgmt Class."

```
define mgmtclass prog1 summer mclass1
spacemgtechnique=automatic migdestination=smpool
description="technical support mgmt class"
```

## Related commands

Table 91. Commands related to *DEFINE MGMTCLASS*

| Command             | Description  |
|---------------------|--|
| ASSIGN DEFMGMTCLASS | Assigns a management class as the default for a specified policy set.                      |
| COPY MGMTCLASS      | Creates a copy of a management class.  |
| DEFINE COPYGROUP    | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE POLICYSET    | Defines a policy set within the specified policy domain.                                   |
| DELETE MGMTCLASS    | Deletes a management class and its copy groups from a policy domain and policy set.        |
| QUERY COPYGROUP     | Displays the attributes of a copy group.   |
| QUERY MGMTCLASS     | Displays information about management classes.   |
| QUERY POLICYSET     | Displays information about policy sets.  |
| UPDATE COPYGROUP    | Changes one or more attributes of a copy group.  |
| UPDATE MGMTCLASS    | Changes the attributes of a management class.  |

### DEFINE NODEGROUP (Define a node group)

Use this command to define a node group. A *node group* is a group of client nodes that are acted upon as if they were a single entity. A node can be a member of one or more node groups.

#### Privilege class

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

#### Syntax

```
►►—DEfIne NODEGroup—group_name—DESCRiption==—description—◄◄
```

#### Parameters

*group\_name*

Specifies the name of the node group that you want to create. The maximum length of the name is 30 characters. The specified name may not be the same as any existing client node name.

DESCRiption

Specifies a description of the node group. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

#### Example: Define a node group

Define a node group named group1.

```
define nodegroup group1
```

#### Related commands

Table 92. Commands related to DEFINE NODEGROUP

| Command                | Description   |
|------------------------|---|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUPMEMBER | Adds a client node to a node group.                     |
| DELETE BACKUPSET       | Deletes a backup set.                                   |
| DELETE NODEGROUP       | Deletes a node group.                                   |
| DELETE NODEGROUPMEMBER | Deletes a client node from a node group.                |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.              |
| QUERY BACKUPSET        | Displays backup sets.                                   |
| QUERY NODEGROUP        | Displays information about node groups.                 |
| UPDATE BACKUPSET       | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP       | Updates the description of a node group.                |

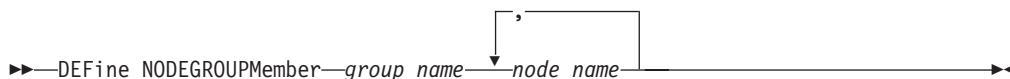
## DEFINE NODEGROUPMEMBER (Define node group member)

Use this command to add a client node to a node group. A *node group* is a group of client nodes that are acted upon as if they were a single entity.

### Privilege class

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

### Syntax



### Parameters

*group\_name*

Specifies the name of the node group to which you want to add a client node.

*node\_name*

Specifies the name of the client node that you want to add to the node group. You can specify one or more names. Separate multiple names with commas; do not use intervening spaces. You can also use wildcard characters when specifying multiple names.

### Example: Define node group members

Define two members, node1 and node2, to a node group, group1.

```
define nodegroupmember group1 node1,node2
```

### Related commands

Table 93. Commands related to DEFINE NODEGROUPMEMBER

| Command                | Description   |
|------------------------|---|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP       | Defines a group of nodes.                               |
| DELETE BACKUPSET       | Deletes a backup set.                                   |
| DELETE NODEGROUP       | Deletes a node group.                                   |
| DELETE NODEGROUPMEMBER | Deletes a client node from a node group.                |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.              |
| QUERY BACKUPSET        | Displays backup sets.                                   |
| QUERY NODEGROUP        | Displays information about node groups.                 |
| UPDATE BACKUPSET       | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP       | Updates the description of a node group.                |

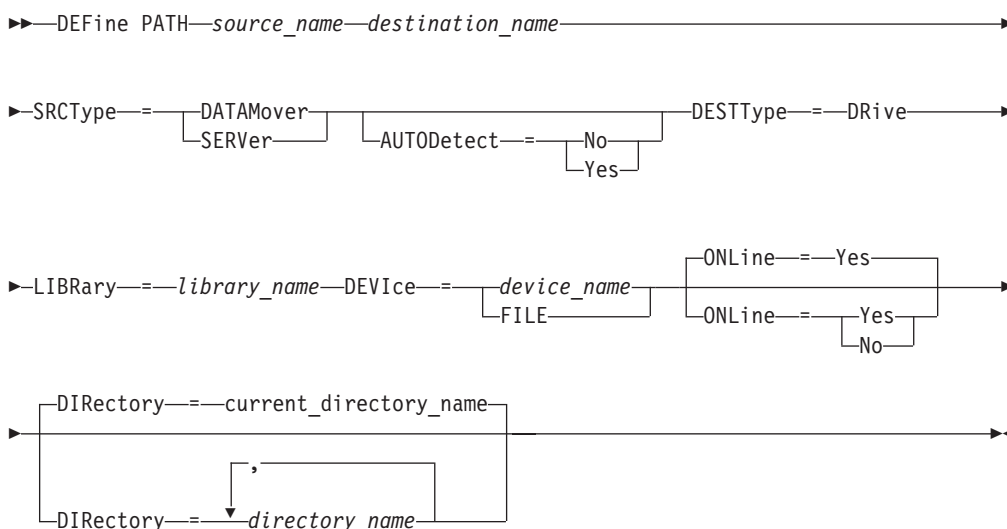
## DEFINE PATH (Define a path)

Use this command to define a path from a source to a destination. A path provides access to a destination from a source. You must define the source and destination before you can define a path. For example, if a path is required between a server and a drive, you must first issue the DEFINE DRIVE command and then issue the DEFINE PATH command. A path must be defined after you issue the DEFINE DRIVE command in order to make the drive usable by IBM Tivoli Storage Manager software.

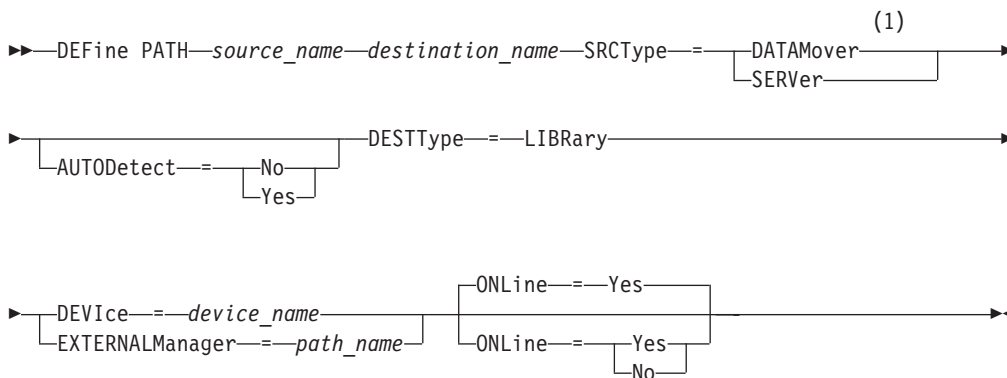
### Privilege class

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

### Syntax when the destination is a drive



### Syntax when the destination is a library



### Notes:

- 1 DATAMOVER only applies to NAS devices.

## Parameters

### *source\_name* (Required)

Specifies the name of source for the path. This parameter is required.

### *destination\_name* (Required)

Specifies the name of the destination. This parameter is required.

The destination can be a drive or a library.

**Attention:** To define a path from a NAS data mover to a library, the library must have LIBTYPE of SCSI, 349x, or ACSLS.

### SRCType (Required)

Specifies the type of the source. This parameter is required. Possible values are:

#### DATAMover

Specifies that a data mover is the source.

#### SERVer

Specifies that a storage agent is the source.

### AUTODetect

Specifies whether the serial number for a drive or library will be automatically updated in the database at the time that the path is defined. This parameter is optional. This parameter is only valid for paths defined from the local server to a drive or a library. Possible values are:

#### No

Specifies that the serial number will not be automatically updated. The serial number is still compared with what is already in the database for the device. The server issues a message if there is a mismatch.

#### Yes

Specifies that the serial number will be automatically updated to reflect the same serial number that the drive reports to IBM Tivoli Storage Manager.

#### Important:

1. If you did not set the serial number when you defined the drive or the library, the server always tries to detect the serial number, and AUTODETECT defaults to YES. If you have previously entered a serial number, then AUTODETECT defaults to NO.
2. The use of AUTODETECT=YES in this command means that the serial number set in the drive or library definition is updated with the detected serial number.
3. **DESTTYPE=DRIVE only:** If you set DESTTYPE=DRIVE and AUTODETECT=YES, then the drive element number in the IBM Tivoli Storage Manager database will be automatically changed to reflect the same element number that corresponds to the serial number of that drive. This is true for drives in a SCSI library. For more information about the element number, see DEFINE DRIVE.
4. Depending on the capabilities of the device, the AUTODETECT parameter may not be supported.

### DESTType (Required)

Specifies the type of the destination. This parameter is required. Possible values are:

## DEFINE PATH

### DRive

Specifies that a drive is the destination. When the destination is a drive, you must specify a library name.

### LIBRARY

Specifies that a library is the destination.

### LIBRARY

Specifies the name of the library to which the drive is assigned. This parameter is required if DESTTYPE=DRIVE. The library and its drives must already be defined to the IBM Tivoli Storage Manager server. If the path is from a NAS data mover to a library, the library must have LIBTYPE of SCSI, 349x, or ACSLS.

### DEVICE

Specifies the name of the device as known to the source, or FILE if the device is a logical drive in a FILE library.

The source uses the device name to access the drive or library. See Table 94 for examples.

Table 94. Examples of device names

| Source to destination  | Example   |
|--|---|
| Server to a drive (not a FILE drive)   | /dev/mt3  |
| Server to a library  | /dev/lb4  |
| Storage agent (on a Windows system) to a drive (not a FILE drive)            | mt3   |
| Storage agent to a drive when the drive is a logical drive in a FILE library | FILE  |
| NAS data mover to a library  | mc0   |
| NAS data mover to a drive  | NetApp NAS file server: rst01<br><br>EMC Celerra NAS file server:<br>c436t011<br><br>IBM System Storage N Series: rst01 |

### Important:

- For more complete information about device names when the source is a server, see the *Administrator's Guide*.
- For information about the device name when the source is a storage agent, see the *Storage Agent User's Guide*.
- For 349X libraries, the alias name is a symbolic name that is specified in the /etc/ibmatl.conf file. For more information, refer to the *IBM Tape Device Drivers Installation and User's Guide*. The Guides can be downloaded from the FTP site at ftp://ftp.software.ibm.com/storage/devdrv/. They are located in the Doc folder.
- For information about how to obtain names for devices that are connected to a NAS file server, consult the product information for the file server. For example, for a NetApp file server, connect to the file server using Telnet and issue the SYSCONFIG command. Use this command to determine device names for drives:  
  
sysconfig -t

Use this command to determine the device name for a library:



```
sysconfig -m
```

## EXTERNALManager

Specifies the location of the external library manager where IBM Tivoli Storage Manager can send media access requests. Use single quotation marks around the value of this parameter. For example, enter: 'c:\Program Files\bin\elm.exe'

This parameter is required when the library name is an external library.

## ONLine

Specifies whether the path is available for use. This parameter is optional. The default is YES. Possible values are:

### Yes

Specifies that the path is available for use.

### No

Specifies that the path is not available for use.

The source and the destination must both be available to use the path.

For example, if the path from a data mover to a drive is online, but either the data mover or the drive is offline, you cannot use the path.

**Attention:** If the path to a library is offline, the server will not be able to access the library. If the server is halted and restarted while the path to the library is offline, the library will not be initialized. See the *Administrator's Guide* for additional information.

## DIRectory

Specifies the directory location or locations where the storage agent reads and writes the files that represent storage volumes for the FILE device class that is associated with the FILE library. The DIRECTORY parameter is also used for devices of type REMOVABLEFILE. For REMOVABLEFILE devices, the DIRECTORY parameter provides information for the server (not a storage agent) along with the DRIVE parameter to describe access to the device. This parameter is optional.

On a storage agent, this parameter is only valid when *all* of the following conditions are true:

- The source type is SERVER (meaning a storage agent that has been defined as a server to this server).
- The source name is the name of a storage agent, *not* the server.
- The destination is a logical drive that is part of a FILE library created when the device class was defined.

If you specified multiple directories for the device class associated with the FILE library, you must specify the same number of directories for each path to the FILE library. Do not change or move existing directories on the server that the storage agent is using so that the device class and the path remain synchronized. Adding directories is permitted. Specifying a mismatched number of directories can cause a run-time failure. See the following example.

The default value for DIRECTORY is the directory of the server at the time the command is issued. The Windows registry is used to locate the default value.

Use a naming convention which you can use to associate the directory with a particular physical drive. This can help ensure that your configuration is valid for sharing the FILE library between the server and storage agent. If the storage agent is on a Windows system, use a universal naming convention

## DEFINE PATH

(UNC) name. When the storage agent lacks permission to access remote storage, the storage agent will experience mount failures.

### Attention:

1. Storage agents access FILE volumes by replacing a directory name in a volume name with a directory name from a directory in the list provided with the DEFINE PATH command. Directories specified with this parameter are not validated on the IBM Tivoli Storage Manager server.
2. IBM Tivoli Storage Manager does not create shares or permissions, or mount the target file system. You must perform these actions before starting the storage agent.

The following illustrates the importance of matching device classes and paths to ensure that storage agents can access newly created FILE volumes.

Suppose you want to use these three directories for a FILE library:

- /opt/tivoli1
- /opt/tivoli2
- /opt/tivoli3

1. You use the following command to set up a FILE library named CLASSA with one drive named CLASSA1 on SERVER1:

```
define devclass classa devtype=file
directory="/opt/tivoli1,/opt/tivoli2,/opt/tivoli3"
shared=yes mountlimit=1
```

2. You want the storage agent STA1 to be able to use the FILE library, so you define the following path for storage agent STA1:

```
define path server1 sta1 srctype=server desttype=drive device=file
directory="/opt/ibm1,/opt/ibm2,/opt/ibm3" library=classa
```

In this scenario, the storage agent, STA1, will replace the directory name /opt/tivoli1 with the directory name /opt/ibm1/ to access FILE volumes that are in the /opt/tivoli1 directory on the server.

3. If file volume /opt/tivoli1/file1.dsm is created on SERVER1, and if the following command is issued,

```
update devclass classa directory="/opt/otherdir,/opt/tivoli2,
/opt/tivoli3"
```

SERVER1 will still be able to access file volume /opt/tivoli1/file1.dsm, but the storage agent STA1 will not be able to access it because a matching directory name in the PATH directory list no longer exists. If a directory name is not available in the directory list associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume will still be accessible from the Tivoli Storage Manager server for reading, failure of the storage agent to access the FILE volume can cause operations to be retried on a LAN-only path or to fail.

### Example: Define a path from a server to a drive

Define a path from a server to a drive. In this case, the server name is *NET1*, the drive name is *TAPEDRV6*, the library is *NETLIB*, and the device name is *mt4*. Set AUTODETECT to NO.

```
define path net1 tapedrv6 srctype=server autodetect=no desttype=drive
library=netlib device=mt4
```

### Example: Define a path from a data mover server to a drive for backup and restore

Define a path from the data mover that is a NAS file server to the drive that the NAS file server will use for backup and restore operations. In this example, the NAS data mover is *NAS1*, the drive name is *TAPEDRV3*, the library is *NASLIB*, and the device name for the drive is *rst01*.

```
define path nas1 tapedrv3 srctype=datamover desttype=drive library=naslib
device=rst01
```

### Example: Define a path from a storage agent to a drive for backup and restore

Define a path from storage agent *SA1* to the drive that the storage agent uses for backup and restore operations. In this example, the library is *TSMLIB*, the drive is *TAPEDRV4*, and the device name for the drive is */dev/mt3*.

```
define path sa1 tapedrv4 srctype=server desttype=drive library=tsmlib
device=/dev/mt3
```

### Example: Define a path to give a storage agent access to shared disk storage

Define a path that gives the storage agent access to files on disk storage shared with the IBM Tivoli Storage Manager server. Drive *FILE9* is defined to library *FILE1* on the server. The storage agent *SA1* accesses *FILE9*. On the storage agent, this data is on directory *\\192.168.1.10\filedata*.

The data for *FILE9* resides on the server at */tsmdata/filedata*.

```
define path sa1 file9 srctype=server desttype=drive library=file1 device=file
directory="\\192.168.1.10\filedata"
```

## Related commands

Table 95. Commands related to *DEFINE PATH*

| Command          | Description   |
|------------------|---|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server.      |
| DEFINE DRIVE     | Assigns a drive to a library.                                       |
| DEFINE LIBRARY   | Defines an automated or manual library.                             |
| DELETE PATH      | Deletes a path from a source to a destination.                      |
| QUERY PATH       | Displays information about the path from a source to a destination. |
| UPDATE DATAMOVER | Changes the definition for a data mover.                            |
| UPDATE PATH      | Changes the attributes associated with a path.                      |

### DEFINE POLICYSET (Define a policy set)

Use this command to define a policy set in a policy domain. A policy set contains management classes, which contain copy groups. You can define one or more policy sets for each policy domain.

To put a policy set into effect, you must activate the policy set by using the **ACTIVATE POLICYSET** command. Only one policy set can be active in a policy domain. The copy groups and management classes within the active policy set determine the rules by which client nodes perform backup, archive, and space management operations, and how the client files stored are managed.

Use the **VALIDATE POLICYSET** command to verify that a policy set is complete and valid before activating it with the **ACTIVATE 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 policy set belongs.

#### Syntax

```

>> DEFINE Policyset—domain_name—policy_set_name—————>
|
| DESCRIPTION==description|
|—————>>

```

#### Parameters

##### *domain\_name* (Required)

Specifies the name of the policy domain to which the policy set belongs.

##### *policy\_set\_name* (Required)

Specifies the name of the policy set. The maximum length of this name is 30 characters. You cannot define a policy set named ACTIVE.

##### DESCRIPTION

Specifies a description for the new policy set. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

#### Example: Define a policy set

Define a policy set called SUMMER for the PROG1 policy domain and include the description, "Programming Group Policies."

```

define policyset prog1 summer
description="Programming Group Policies"

```

#### Related commands

Table 96. Commands related to **DEFINE POLICYSET**

| Command            | Description                           |
|--------------------|---------------------------------------|
| ACTIVATE POLICYSET | Validates and activates a policy set. |
| COPY MGMTCLASS     | Creates a copy of a management class. |
| COPY POLICYSET     | Creates a copy of a policy set.       |

Table 96. Commands related to *DEFINE POLICYSET* (continued)

| Command            | Description  |
|--------------------|--|
| DEFINE DOMAIN      | Defines a policy domain that clients can be assigned to.   |
| DEFINE MGMTCLASS   | Defines a management class.  |
| DELETE POLICYSET   | Deletes a policy set, including its management classes and copy groups, from a policy domain.        |
| QUERY POLICYSET    | Displays information about policy sets.  |
| UPDATE POLICYSET   | Changes the description of a policy set.   |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

### DEFINE PROFASSOCIATION (Define a profile association)

Use this command on a configuration manager to associate one or more objects with a configuration profile for distribution to subscribing managed servers. After a managed server subscribes to a profile, the configuration manager sends object definitions associated with the profile to the managed server where they are stored in the database. Objects created this way in the database of a managed server become managed objects. An object can be associated with more than one profile.

You can use this command to define an initial set of profile associations and to add to existing associations.

You can associate the following types of objects with a profile:

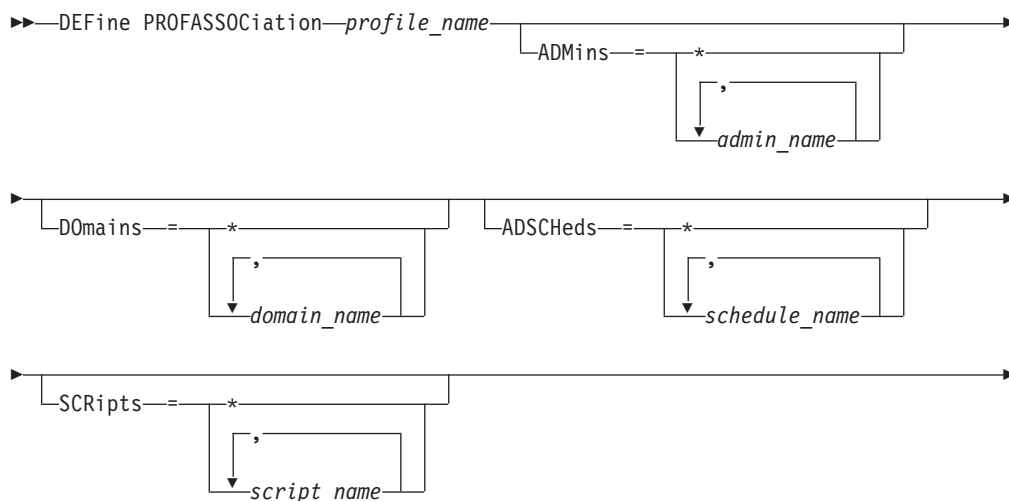
- Administrator registrations and authorities
- Policy domains, which include the domains' policy sets, management classes, copy groups, and client schedules
- Administrative schedules
- Server command scripts
- Client option sets
- Server definitions
- Server group definitions

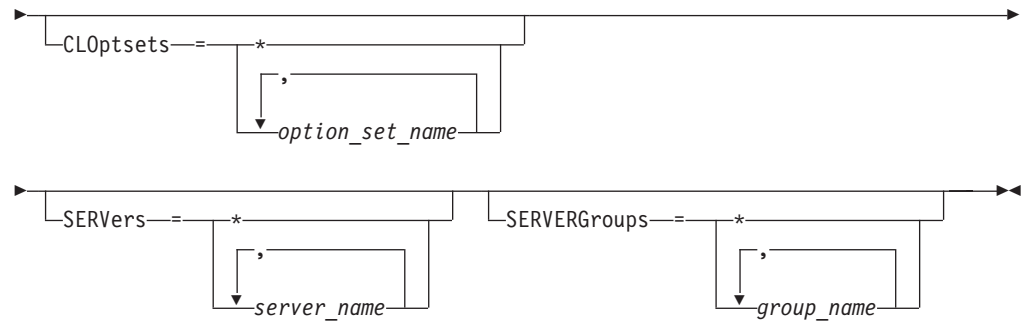
**Tip:** The configuration manager does not distribute status information for an object to managed servers. For example, information such as the number of days since an administrator last accessed the server is not distributed to managed servers. This type of information is maintained in the databases of the individual managed servers.

### Privilege class

To issue this command, you must have system privilege.

### Syntax





## Parameters

### *profile\_name* (Required)

Specifies the name of the configuration profile.

### ADMinS

Specifies administrators to associate with the profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all administrators that are registered with the configuration manager. If you specify the match-all definition and later add more administrators, they are automatically distributed through the profile.

The configuration manager distributes the administrator name, password, contact information, and authorities of administrators associated with the profile. The configuration manager does not distribute the following:

- The administrator named SERVER\_CONSOLE, even if you use a match-all definition
- The locked or unlocked status of an administrator

When the profile already has administrators associated with it, the following apply:

- If you specify a list of administrators and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you specify a match-all definition and a list of administrators already exists, Tivoli Storage Manager replaces the list with the match-all definition.
- If you specify a list of administrators, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the ADMINs=\* parameter.

### DOmainS

Specifies policy domains to associate with the profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all domains that are defined on the configuration manager. If you specify the match-all definition and later add more domains, they are automatically distributed through the profile.

The configuration manager distributes domain information that includes definitions of policy domains, policy sets, management classes, copy groups, and client schedules. The configuration manager does not distribute the ACTIVE policy set. Administrators on a managed server can activate any policy set within a managed domain on a managed server.

## DEFINE PROFASSOCIATION

When the profile already has domains associated with it, the following apply:

- If you specify a list of domains and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you use a match-all definition and a list of domains already exists, Tivoli Storage Manager replaces the list with the match-all definition.
- If you specify a list of domains, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the DOMAINS=\* parameter.

**Important:** Client operations such as backup and archive fail if destination pools do not exist. Therefore, managed servers that subscribe to this profile must have definitions for any storage pools specified as destinations in the associated domains. Use the RENAME STGPOOL command to rename existing storage pools to match the destination names distributed.

### ADScheds

Specifies administrative schedules to associate with the profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all administrative schedules that are defined on the configuration manager. If you specify the match-all definition and later add more administrative schedules, they are automatically distributed through the profile.

**Tip:** Administrative schedules are not active when they are distributed by a configuration manager. An administrator on a managed server must activate any schedule to have it run on that server.

When the profile already has administrative schedules associated with it, the following apply:

- If you specify a list of administrative schedules and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you use a match-all definition and a list of administrative schedules already exists, Tivoli Storage Manager replaces the list with the match-all definition.
- If you specify a list of administrative schedules, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the ADSCHEDS=\* parameter.

### SCRipts

Specifies server command scripts to associate with the profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all scripts that are defined on the configuration manager. If you specify the match-all definition and later add more scripts, they are automatically distributed through the profile.

When the profile already has scripts associated with it, the following apply:

- If you specify a list of scripts and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you use a match-all definition and a list of scripts already exists, Tivoli Storage Manager replaces the list with the match-all definition.



- If you specify a list of scripts, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the SCRIPTS=\* parameter.

**CLOptsets**

Specifies client option sets to associate with the profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all client option sets that are defined on the configuration manager. If you specify the match-all definition and later add more client option sets, they are automatically distributed through the profile.

When the profile already has client option sets associated with it, the following apply:

- If you specify a list of client option sets and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you use a match-all definition and a list of client option sets already exists, Tivoli Storage Manager replaces the list with the match-all definition.
- If you specify a list of client option sets, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the CLOPSETS=\* parameter.

**SERVers**

Specifies server definitions to associate with the profile. The definitions are distributed to managed servers that subscribe to this profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all servers that are defined on the configuration manager. If you specify the match-all definition and later add more servers, they are automatically distributed through the profile.

The configuration manager distributes the following server attributes: communication method, IP address, port address, server password, URL, and the description. Distributed server definitions always have the ALLOWREPLACE attribute set to YES on the managed server, regardless of this parameter's value on the configuration manager. On the managed server, you can use the UPDATE SERVER command to set all other attributes.

When the profile already has servers associated with it, the following apply:

- If you specify a list of servers and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you use a match-all definition and a list of servers already exists, Tivoli Storage Manager replaces the list with the match-all definition.
- If you specify a list of servers, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the SERVERS=\* parameter.

**Note:**

1. A server definition on a managed server is not replaced by a definition from the configuration manager unless you have allowed replacement of the definition on the managed server. To allow replacement, on the

## DEFINE PROFASSOCIATION

managed server update the server definition by using the UPDATE SERVER command with ALLOWREPLACE=YES.

2. If a configuration manager distributes a server definition to a managed server, and a server group of the same name exists on the managed server, the distributed server definition replaces the server group definition.

### SERVERGroups

Specifies server groups to associate with the profile. You can use wildcard characters in the names. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all definition, an asterisk (\*) by itself, to specify all server groups that are defined on the configuration manager. If you specify the match-all definition and later add more server groups, they are automatically distributed through the profile.

**Tip:** A configuration manager does not distribute a server group definition to a managed server if the managed server has a server defined with the same name as that of the server group.

When the profile already has server groups associated with it, the following apply:

- If you specify a list of server groups and a list already exists, Tivoli Storage Manager combines the new list with the existing list.
- If you use a match-all definition and a list of server groups already exists, Tivoli Storage Manager replaces the list with the match-all definition.
- If you specify a list of server groups, and a match-all definition had previously been specified, Tivoli Storage Manager ignores the list. To remove the match-all definition, issue the DELETE PROFASSOCIATION command with the SERVERGROUPS=\* parameter.

### Example: Associate a specific domain with a specific profile

Associate a domain named MARKETING with a profile named DELTA.

```
define profassociation delta domains=marketing
```

### Example: Associate all domains with a specific profile

You have already associated a list of domains with a profile named GAMMA. Now associate all domains defined on the configuration manager with the profile.

```
define profassociation gamma domains=*
```

### Related commands

Table 97. Commands related to DEFINE PROFASSOCIATION

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                       |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                    |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                  |

Table 97. Commands related to DEFINE PROFASSOCIATION (continued)

| Command            | Description  |
|--------------------|--|
| NOTIFY SUBSCRIBERS | Notifies servers to refresh their configuration information.   |
| QUERY PROFILE      | Displays information about configuration profiles.             |
| SET CONFIGMANAGER  | Specifies whether a server is a configuration manager.         |
| UNLOCK PROFILE     | Enables a locked profile to be distributed to managed servers. |
| UPDATE PROFILE     | Changes the description of a profile.                          |

### DEFINE PROFILE (Define a profile)

Use this command on a configuration manager to define a profile (a set of configuration information) that can be distributed to managed servers.

After defining a profile, you can use the DEFINE PROFASSOCIATION command to specify objects to be distributed to managed servers subscribing to the profile.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

```
►►—DEfIne PROFIle—profile_name—[DESCRiption—=description—]—►►
```

#### Parameters

##### *profile\_name* (Required)

Specifies the name of the profile. The maximum length of the name is 30 characters.

##### DESCRiption

Specifies a description of the profile. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. This parameter is optional.

#### Example: Define a new profile

Define a profile named ALPHA with a description of "Programming Center".

```
define profile alpha
description="Programming Center"
```

#### Related commands

Table 98. Commands related to DEFINE PROFILE

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                   |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                             |
| DEFINE SUBSCRIPTION    | Subscribes a managed server to a profile.                      |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.           |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                |
| LOCK PROFILE           | Prevents distribution of a configuration profile.              |
| QUERY PROFILE          | Displays information about configuration profiles.             |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.         |
| UNLOCK PROFILE         | Enables a locked profile to be distributed to managed servers. |

Table 98. Commands related to *DEFINE PROFILE* (continued)

| Command        | Description                           |
|----------------|---------------------------------------|
| UPDATE PROFILE | Changes the description of a profile. |

## DEFINE RECMEDMACHASSOCIATION (Associate recovery media with a machine)

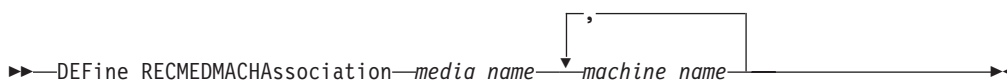
Use this command to associate recovery media with one or more machines. A machine is associated with recovery media so that the location of the boot media and its list of volume names are available to recover the machine. To retrieve the information, issue the QUERY MACHINE command. This information will be included in the plan file to help you recover the client machines.

To associate a machine with recovery media, both the machine and media must be defined to Tivoli Storage Manager. A machine remains associated with the media until the association, the media, or the machine is deleted.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *media\_name* (Required)

Specifies the name of the recovery media with which one or more machines will be associated.

#### *machine\_name* (Required)

Specifies the name of the machines to be associated with the recovery media. A machine can be associated with multiple recovery media. To specify a list of machines, separate the names with commas and no intervening spaces. You can use wildcard characters to specify a name.

### Example: Associate machines to recovery media

Associate machines DISTRICT1 and DISTRICT5 to the DIST5RM recovery media.

```

define recmedmachassociation dist5rm
district1,district5
  
```

### Related commands

Table 99. Commands related to DEFINE RECMEDMACHASSOCIATION

| Command                      | Description   |
|------------------------------|---|
| DEFINE MACHINE               | Defines a machine for DRM.                                |
| DEFINE RECOVERYMEDIA         | Defines the media required to recover a machine.          |
| DELETE MACHINE               | Deletes a machine.  |
| DELETE RECMEDMACHASSOCIATION | Deletes association between recovery media and a machine. |
| DELETE RECOVERYMEDIA         | Deletes recovery media.                                   |
| QUERY MACHINE                | Displays information about machines.                      |

*Table 99. Commands related to DEFINE RECMEDMACHASSOCIATION (continued)*

| Command             | Description                                    |
|---------------------|--|
| QUERY RECOVERYMEDIA | Displays media available for machine recovery. |

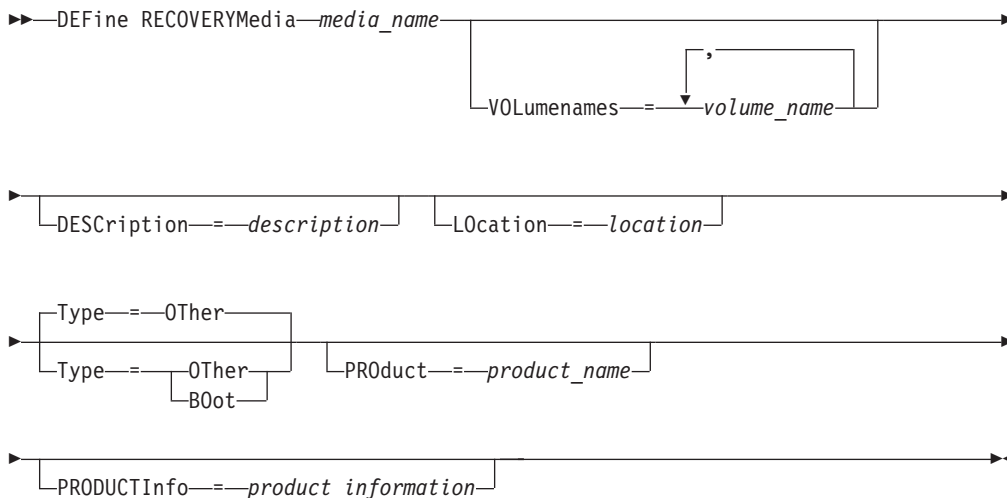
## DEFINE RECOVERYMEDIA (Define recovery media)

Use this command to define the media needed to recover a machine. The same media can be associated with multiple machines. To display the information, use the QUERY MACHINE command. This information will be included in the plan file to help you to recover the client machines.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *media\_name* (Required)

Specifies the name of the recovery media to be defined. The name can be up to 30 characters.

#### **VOLumenames**

Specifies the names of volumes that contain the recoverable data (for example, operating system image copies). This parameter is required if you specify a media type of `BOOT`. Specify boot media volume names in the order in which they are to be inserted into the machine at recovery time. The maximum length of the volume names list is 255 characters. Enclose the list in quotation marks if it contains any blank characters.

#### **DESCription**

Specifies the description of the recovery media. This parameter is optional. The maximum length is 255 characters. Enclose the text in quotation marks if it contains any blank characters.

#### **LLocation**

Specifies the location of the recovery media. This parameter is optional. The maximum length is 255 characters. Enclose the text in quotation marks if it contains any blank characters.

#### **Type**

Specifies the type of recovery media. This parameter is optional. The default is `OTHER`.



## BOot

Specifies that this is boot media. You must specify volume names if the type is BOOT.

## OTher

Specifies that this is not boot media. For example, a CD that contains operating system manuals.

## PROduct

Specifies the name of the product that wrote to this media. This parameter is optional. The maximum length is 16 characters. Enclose the text in quotation marks if it contains any blank characters.

## PRODUCTInfo

Specifies information about the product that wrote to the media. This would be information that you may need to restore the machine. This parameter is optional. The maximum length is 255 characters. Enclose the text in quotation marks if it contains any blank characters.

## Example: Define the media needed to recover a machine

Define the recovery media named DIST5RM. Include a description and the location.

```
define recoverymedia dist5rm
description="district 5 base system image"
location="district 1 vault"
```

## Related commands

Table 100. Commands related to DEFINE RECOVERYMEDIA

| Command                      | Description                                    |
|------------------------------|--|
| DEFINE RECMEDMACHASSOCIATION | Associates recovery media with a machine.      |
| DELETE RECOVERYMEDIA         | Deletes recovery media.                        |
| QUERY RECOVERYMEDIA          | Displays media available for machine recovery. |
| UPDATE RECOVERYMEDIA         | Changes the attributes of recovery media.      |

### DEFINE SCHEDULE (Define a client or an administrative command schedule)

Use this command to create a client or administrative command schedule.

The DEFINE SCHEDULE command takes two forms: one if the schedule applies to client operations, one if the schedule applies to administrative commands. Within these two forms, you can select either classic or enhanced style schedules. The syntax and parameters for each form are defined separately.

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 processing 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. Options associated with each schedule style (classic and enhanced ) determine when the startup windows should begin.

*Table 101. Commands related to DEFINE SCHEDULE*

| Command              | Description   |
|----------------------|---|
| COPY SCHEDULE        | Creates a copy of a schedule.   |
| DEFINE ASSOCIATION   | Associates clients with a schedule.   |
| DELETE SCHEDULE      | Deletes a schedule from the database.   |
| QUERY EVENT          | Displays information about scheduled and completed events for selected clients.                 |
| QUERY SCHEDULE       | Displays information about schedules.   |
| SET MAXCMDRETRIES    | Specifies the maximum number of retries after a failed attempt to execute a scheduled command.  |
| SET MAXSCHEDSESSIONS | Specifies the maximum number of client/server sessions available for processing scheduled work. |
| SET RETRYPERIOD      | Specifies the time between retry attempts by the client scheduler.                              |
| UPDATE SCHEDULE      | Changes the attributes of a schedule.   |

## DEFINE SCHEDULE (Define a client backup or archive schedule)

Use the DEFINE SCHEDULE command to define a client schedule. Tivoli Storage Manager uses this schedule to perform automatic backup and archive operations for your client workstation at specified intervals or days. After you define a schedule, use the DEFINE ASSOCIATION command to associate the client with the schedule.

You must start the client scheduler on the client workstation for Tivoli Storage Manager to process the schedule.

Not all clients can run all scheduled operations, even though you can define the schedule on the server and associate it with the client. For example, a Macintosh client cannot run a schedule when the action is to restore or retrieve files, or run an executable script. An executable script is also known as a command file, a batch file, or a script on different client operating systems.

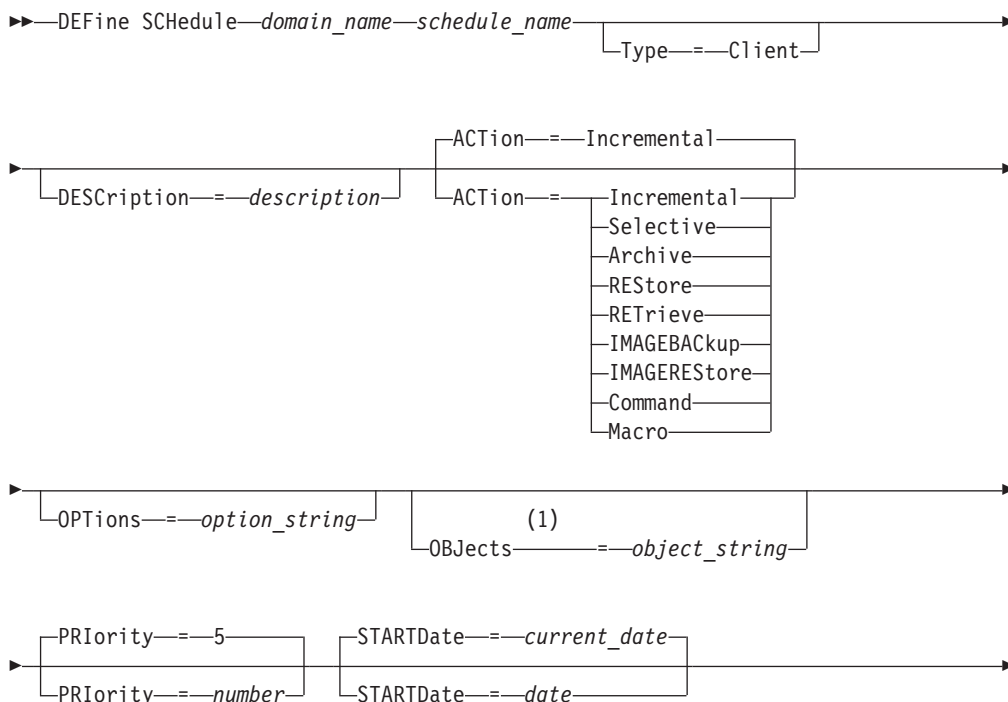
Tivoli Storage Manager cannot run multiple schedules concurrently for the same client node.

### Privilege class

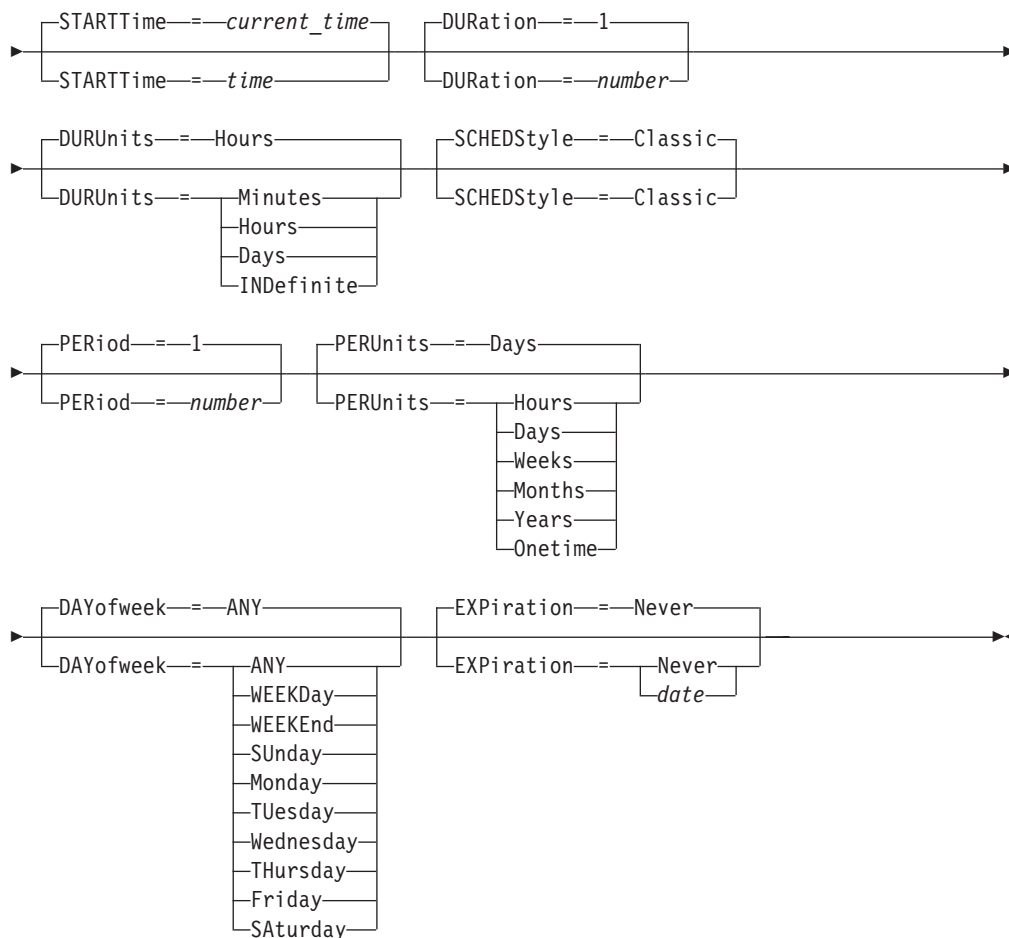
To define a client schedule, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the schedule belongs.

### Syntax

#### Classic client schedule



## DEFINE SCHEDULE

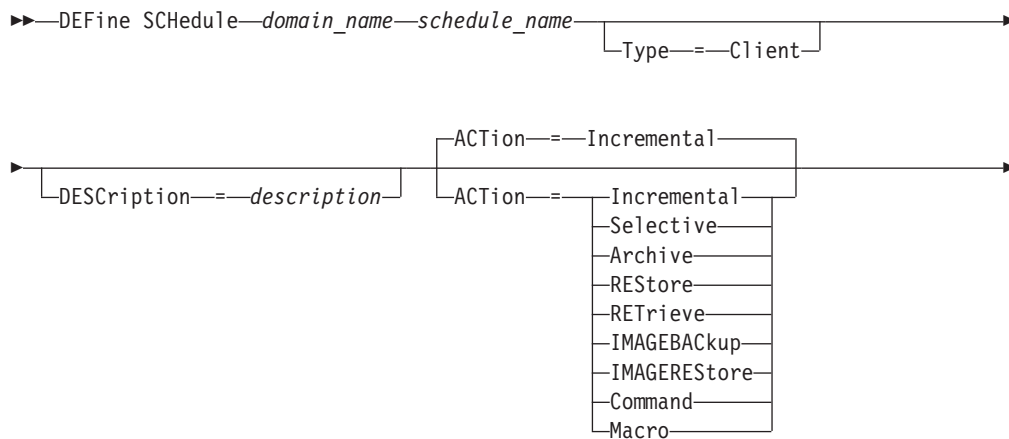


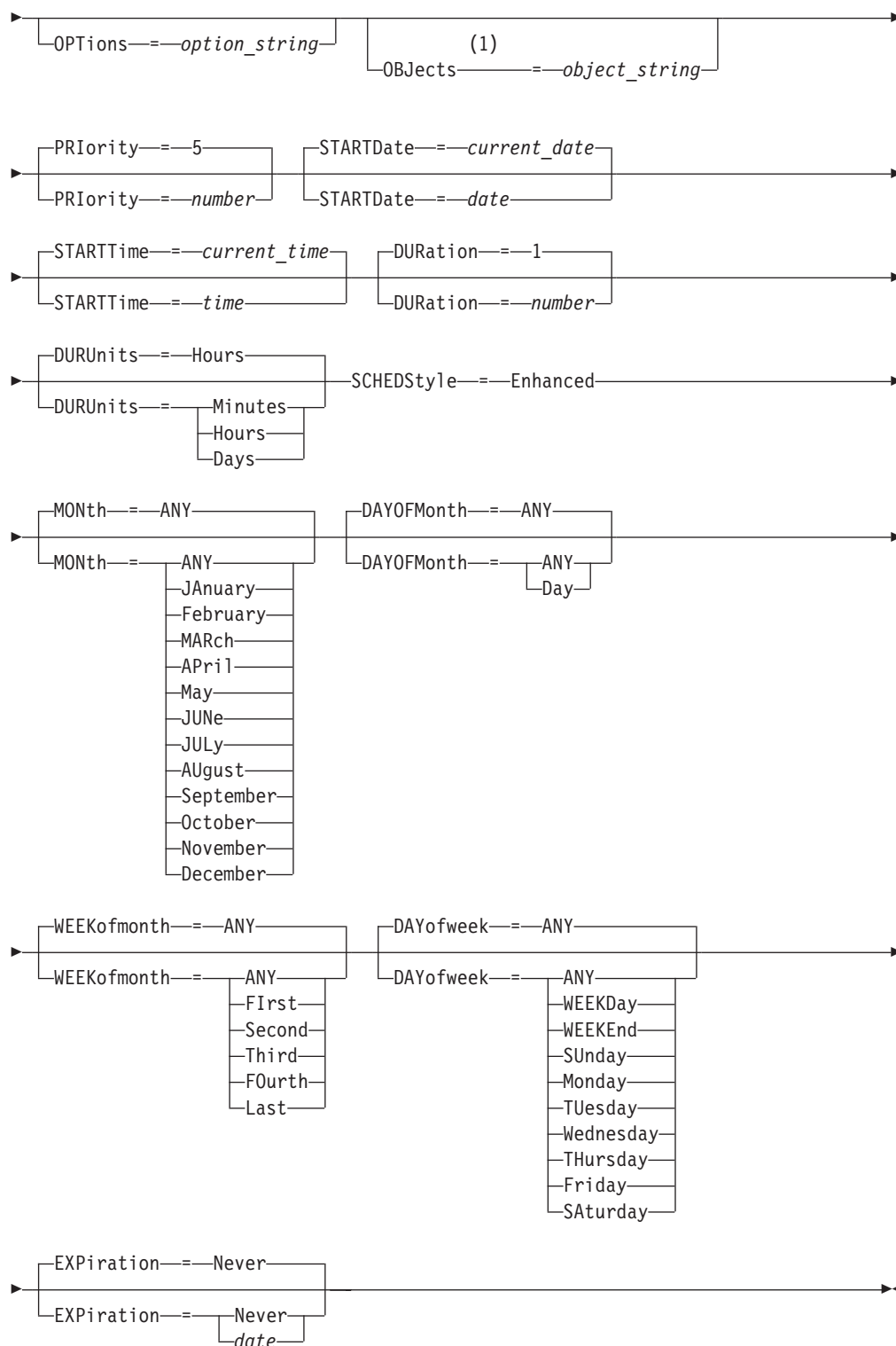
### Notes:

- 1 The OBJECTS parameter is optional when ACTION=INCREMENTAL, but is required for other actions.

### Syntax

#### Enhanced client schedule





## Notes:

- 1 The OBJECTS parameter is optional when ACTION=INCREMENTAL, but is required for other actions.

## DEFINE SCHEDULE

### Parameters

#### *domain\_name* (Required)

Specifies the name of the policy domain to which this schedule belongs.

#### *schedule\_name* (Required)

Specifies the name of the schedule to be defined. You can specify up to 30 characters for the name.

#### **Type=Client**

Specifies that a schedule for a client is defined. This parameter is optional.

#### **DESCription**

Specifies a description of the schedule. This parameter is optional. You can specify up to 255 characters for the description. Enclose the description in quotation marks if it contains any blank characters.

#### **ACTion**

Specifies the action that occurs when this schedule is processed. Possible values are:

##### **Incremental**

Specifies that the schedule backs up all files that are new or that have changed since the last incremental backup. Incremental also backs up any file for which all existing backups may have expired.

##### **Selective**

Specifies that the schedule backs up only files that are specified with the OBJECTS parameter.

##### **Archive**

Specifies that the schedule archives files that are specified with the OBJECTS parameter.

##### **REStore**

Specifies that the schedule restores files that are specified with the OBJECTS parameter.

When you specify ACTION=RESTORE for a scheduled operation, and the REPLACE option is set to PROMPT, no prompting occurs. If you set the option to PROMPT, the files are skipped.

If you specify a second file specification, this second file specification acts as the restore destination. If you need to restore multiple groups of files, schedule one for each file specification that you need to restore.

##### **REtrieve**

Specifies that the schedule retrieves files that are specified with the OBJECTS parameter.

##### **IMAGEBACKup**

Specifies that the schedule backs up logical volumes that are specified with the OBJECTS parameter.

##### **IMAGERESTore**

Specifies that the schedule restores logical volumes that are specified with the OBJECTS parameter.

##### **Command**

Specifies that the schedule processes a client operating system command or script that is specified with the OBJECTS parameter.

## Macro

Specifies that a client processes a macro whose file name is specified with the OBJECTS parameter.

## OPTions

Specifies the client options that you specify to the scheduled command at the time the schedule is processed. This parameter is optional.

Only those options that are valid on the scheduled command may be specified for this parameter. Refer to the appropriate client manual for information about options that are valid from the command line. All options described there as valid only on the initial command line will either result in an error or simply be ignored when the schedule from the server is run. For example, do not include the following options because they have no effect when the client processes the scheduled command:

MAXCMDRETRIES  
OPTFILE  
QUERYSCHEDPERIOD  
RETRYPERIOD  
SCHEDLOGNAME  
SCHEDMODE  
SERVERNAME  
TCPCLIENTADDRESS  
TCPCLIENTPORT

If the option string contains multiple options or options with embedded spaces, surround the entire option string with one pair of single quotes. Individual options that contain spaces should be enclosed in double quotes. A leading minus sign is required in front of the option. Errors can occur if the option string contains spaces that are not quoted correctly.

The following examples show how to specify some client options:

- To specify subdir=yes and domain all-local -systemobject, enter:  
options='-subdir=yes -domain="all-local -c: -systemobject"'
- To specify domain all-local -c: -d:, enter:  
options='-domain="all-local -c: -d:"'

## OBjects

Specifies the objects for which the specified action is performed. Use a single space between each object. This parameter is required except when ACTION=INCREMENTAL. If the action is a backup, archive, retrieve, or restore operation, the objects are file spaces, directories, or logical volumes. See the *Backup-Archive Clients Installation and User's Guide* for command syntax information. If the action is to run a command or macro, the object is the name of the command or macro to run.

When you specify ACTION=INCREMENTAL without specifying a value for this parameter, the scheduled command is invoked without specified objects and attempts to process the objects as defined in the client's option file. To select all file spaces or directories for an action, explicitly list them in the object string. Entering only an asterisk in the object string causes the backup to occur only for the directory where the scheduler was started.

## Important:

## DEFINE SCHEDULE

- If you specify a second file specification, and it is not a valid destination, you receive this error:  
ANS1082E Invalid destination file specification <filespec> entered.
- If you specify more than two file specifications, you receive this error:  
ANS1102E Excessive number of command line arguments passed to the program!

When you specify ACTION=ARCHIVE, INCREMENTAL, or SELECTIVE for this parameter, you can list a maximum of twenty (20) file specifications.

Enclose the object string in double quotes if it contains blank characters (spaces), and then surround the double quotes with single quotes. If the object string contains multiple file names, enclose each file name with its own pair of double quotes, then surround the entire string with one pair of single quotes. Errors can occur if file names contain a space that is not quoted correctly. The following examples show how to specify some file names:

- To specify /home/file 2, /home/gif files, and /home/my test file, enter:  
OBJECTS='"/home/file 2" "/home/gif files" "/home/my test file"'
- To specify /home/test file, enter:  
OBJECTS='"/home/test file"'

### PRIority

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer from 1 to 10, with 1 being the highest priority and 10 being the lowest. The default is 5.

If two or more schedules have the same window start time, the value you specify determines when Tivoli Storage Manager processes the schedule. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

### STARTDate

Specifies the date for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current date. Use this parameter with the STARTTIME parameter to specify when the initial startup window of the schedule starts.

You can specify the date using one of the values below:

| Value               | Description   | Example         |
|---------------------|---|-----------------|
| MM/DD/YYYY          | A specific date   | 09/15/1998      |
| TODAY               | The current date  | TODAY           |
| TODAY+days or +days | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3. |

### STARTTime

Specifies the time for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current time. This parameter is used in conjunction with the STARTDATE parameter to specify when the initial startup window begins.

You can specify the time using one of the values below:

| Value    | Description     | Example  |
|----------|-----------------|----------|
| HH:MM:SS | A specific time | 10:30:08 |



| Value                                | Description  | Example  |
|--------------------------------------|--|--|
| <b>NOW</b>                           | The current time                                   | NOW  |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes specified  | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW+02:00 or STARTTIME=+02:00, the beginning of the startup window is at 7:00. |
| <b>NOW-HH:MM</b><br><b>or -HH:MM</b> | The current time minus hours and minutes specified | NOW-02:00 <b>or</b> -02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW-02:00 or STARTTIME=-02:00, the beginning of the startup window is at 3:00. |

### DURation

Specifies the number of units that define the length of the startup window of the scheduled operation. This parameter is optional. This value must be from 1 to 999. The default is 1.

Use this parameter 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 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.

**Tip:** Define schedules with durations longer than 10 minutes. Doing this will give the Tivoli Storage Manager scheduler enough time to process the schedule and prompt the client.

### DURUnits

Specifies the time units used to determine the duration of the window in which the schedule can start. This parameter is optional. The default is HOURS.

Use this parameter with the DURATION parameter to specify how long the startup window remains open to process 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 schedule may not necessarily complete processing within this window. If the schedule needs to be retried for any reason, the retry attempts must begin before the startup window elapses, or the operation does not restart.

The default value for the length of the startup window is 1 hour. 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 of the scheduled operation has an

## DEFINE SCHEDULE

indefinite duration. The schedule can run any time after the scheduled start time, until the schedule expires. You cannot specify `DURUNITS=INDEFINITE`, unless you specify `PERUNITS=ONETIME`. The `INDEFINITE` value is not allowed with enhanced schedules.

### **SCHEDStyle**

This parameter is optional. `SCHEDSTYLE` defines either the interval between times when a schedule should run, or the days on which it should run. The default is the classic syntax.

Possible values are:

#### **Classic**

The parameters for the Classic syntax are: `PERIOD`, `PERUNITS`, and `DAYOFWEEK`. You cannot use these parameters: `MONTH`, `DAYOFMONTH`, and `WEEKOFMONTH`.

#### **Enhanced**

The parameters for the Enhanced syntax are: `MONTH`, `DAYOFMONTH`, `WEEKOFMONTH`, and `DAYOFWEEK`. You cannot use these parameters: `PERIOD` and `PERUNITS`.

### **PERiod**

Specifies the length of time between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. You can specify an integer from 1 to 999. The default is 1.

Use this parameter 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 five days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default is 1 day.

This value is ignored if you specify `PERUNITS=ONETIME`.

### **PERUnits**

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. The default is `DAYS`.

Use this parameter 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 default is 1 day. Possible values are:

#### **Hours**

Specifies that the time between startup windows is in hours.

#### **Days**

Specifies that the time between startup windows is in days.

#### **Weeks**

Specifies that the time between startup windows is in weeks.

#### **Months**

Specifies that the time between startup windows is in months.

When you specify `PERUNITS=MONTHS`, the scheduled operation will be processed each month on the same date. For example, if the start date for the scheduled operation is 02/04/1998, the schedule will process on the 4th of every month thereafter. However, if the date is not valid for the next

month, then the scheduled operation will be processed on the last valid date in the month. Thereafter, subsequent operations are based on this new date. For example, if the start date is 03/31/1998, the next month's operation will be scheduled for 04/30/1998. Thereafter, all subsequent operations will be on the 30th of the month until February. Because February has only 28 days, the operation will be scheduled for 02/28/1999. Subsequent operations will be processed on the 28th of the month.

### **Years**

Specifies that the time between startup windows for the schedule is in years.

When you specify PERUNITS=YEARS, the scheduled operation will be processed on the same month and date of each year. For example, if the start date for the scheduled operation is 02/29/2004, the next year's scheduled operation will be 02/28/2005 because February only has 28 days. Thereafter, subsequent operations will be scheduled for February 28th.

### **Onetime**

Specifies that the schedule processes once. This value overrides the value you specified for the PERIOD parameter.

### **DAYofweek**

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can specify different options for the DAYofweek parameter, depending on whether the schedule style has been defined as Classic or Enhanced:

#### **Classic Schedule**

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can either specify one day of the week, or WEEKDAY, WEEKEND, or ANY. If the start date and start time fall on a day that does not correspond to a day you specify, the start date and start time 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 processed when you would expect. The default is ANY.

#### **Enhanced Schedule**

Specifies the days of the week on which to run the schedule. You can either specify multiple days separated by commas and no intervening blanks, or WEEKDAY, WEEKEND, or ANY. If you specify multiple days, the schedule will run on each of the specified days. If you specify WEEKDAY or WEEKEND, you must also specify either WEEKOFMONTH=FIRST or WEEKOFMONTH=LAST, and the schedule will run just once per month.

The default value is ANY, meaning the schedule will run every day of the week or on the day or days determined by other enhanced schedule parameters. DAYOFWEEK must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

Possible values for the DAYofweek parameter are:

#### **ANY**

Specifies that the startup window can begin on any day of the week.

## DEFINE SCHEDULE

### **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.

### **Saturday**

Specifies that the startup window begins on Saturday.

### **MONth**

Specifies the months of the year during which to run the schedule. This parameter is used only with enhanced schedules. Specify multiple values by using commas and no intervening blanks. The default value is ANY. This means the schedule will run during every month of the year.

### **DAYOFMonth**

Specifies the day of the month to run the schedule. This parameter is used only with enhanced schedules. You can either specify ANY or a number from -31 through 31, excluding zero. Negative values are a day from the end of the month, counting backwards. For example, the last day of the month is -1, the next-to-the-last day of the month is -2, and so on. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run on each of the specified days of the month. If multiple values resolve to the same day, the schedule will run only once that day.

The default value is ANY. This means the schedule will run on every day of the month or on the days determined by other enhanced schedule parameters. DAYOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFWEEK or WEEKOFMONTH parameters.

### **WEEKofmonth**

Specifies the week of the month in which to run the schedule. This parameter is used only with enhanced schedules. A week is considered any seven-day period which does not start on a particular day of the week. You can specify FIRST, SECOND, THIRD, FOURTH, LAST, or ANY. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run during each of the specified weeks of the month. If multiple values resolve to the same week, the schedule will run only once during that week.

The default value is ANY, meaning the schedule will run during every week of the month or on the day or days determined by other enhanced schedule parameters. WEEKOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

### EXPIRATION

Specifies the date after which this schedule is no longer used. This parameter is optional. The default is NEVER. Possible values are:

#### Never

Specifies that the schedule never expires.

#### *expiration\_date*

Specifies the date on which this schedule expires, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the date you specify.

### Example: Define a schedule for a monthly incremental backup

Define a schedule named MONTHLY\_BACKUP that initiates an incremental backup of all associated nodes. Specify the start date as Tuesday, May 1, 2001. This date does not match the specified day of the week (Sunday), so the initial startup window begins on the first Sunday after May 1, 2001 (05/01/2001). 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.

```
define schedule standard monthly_backup
description="Monthly Backup of c: and d: drives"
objects="c:\* d:\*"
startdate=05/01/2001 starttime=01:00
duration=2 durunits=hours period=1
perunits=months dayofweek=sunday
```

### Example: Define a schedule for a weekly incremental backup

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, June 7, 1997 (06/07/1997), to 03:00 on Sunday, June 8, 1997 (06/08/1997). Subsequent windows begin at 23:00, every Saturday. No messages are returned to the client node when this schedule is run.

```
define schedule employee_records weekly_backup
startdate=06/07/1997 starttime=23:00 duration=4
durunits=hours perunits=weeks
dayofweek=saturday options=-quiet
```

### Example: Define a schedule that archives a specific directory every quarter

Define a schedule that archives specific files quarterly on the last Friday of the month.

```
define schedule employee_records quarterly_archive
starttime=20:00 action=archive
object=/home/employee/records/*
duration=1 durunits=hour schedstyle=enhanced
month=mar,jun,sep,dec weekofmonth=last dayofweek=fri
```

## DEFINE SCHEDULE

### DEFINE SCHEDULE (Define a schedule for an administrative command)

Use the DEFINE SCHEDULE command to create a new schedule for processing an administrative command.

You can include scripts in an administrative command schedule so the commands are processed automatically.

#### Note:

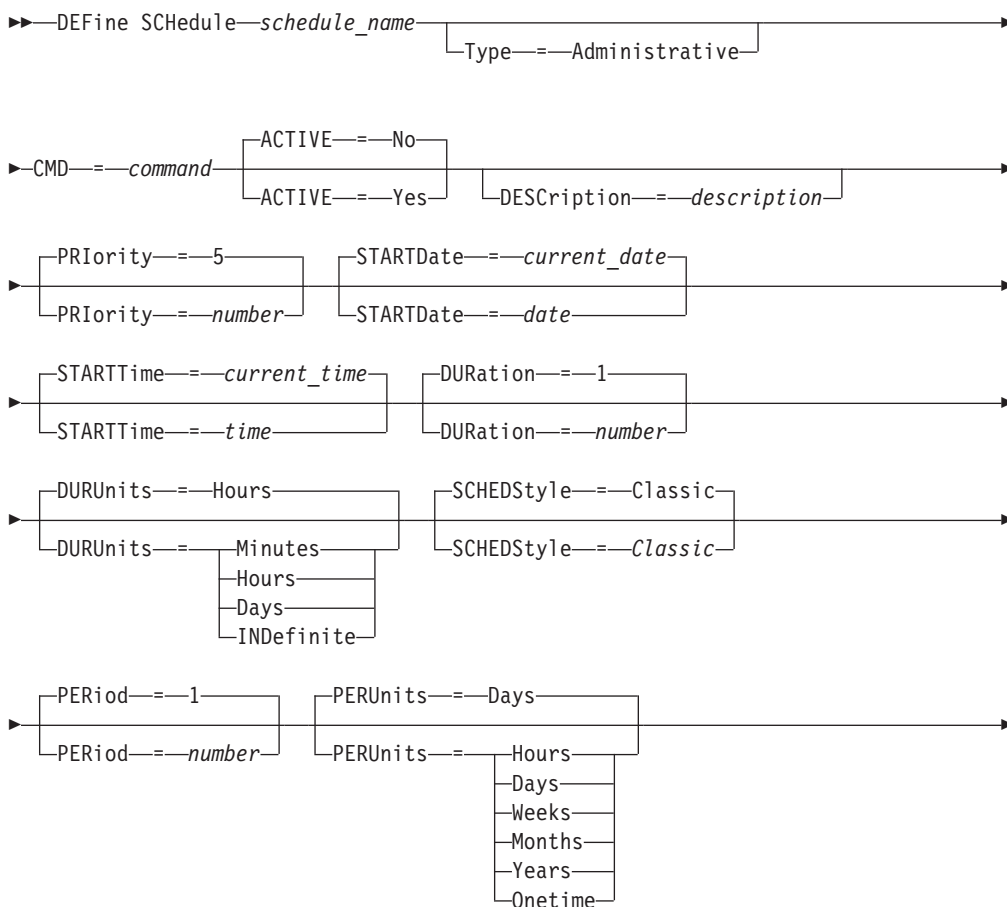
1. You cannot schedule the MACRO command or the QUERY ACTLOG command.
2. If you are scheduling a command that specifies the WAIT parameter, the parameter must be set to YES in order for the process to provide a return code to the session that started it. For more information about the WAIT parameter, see “Server command processing” on page 14.

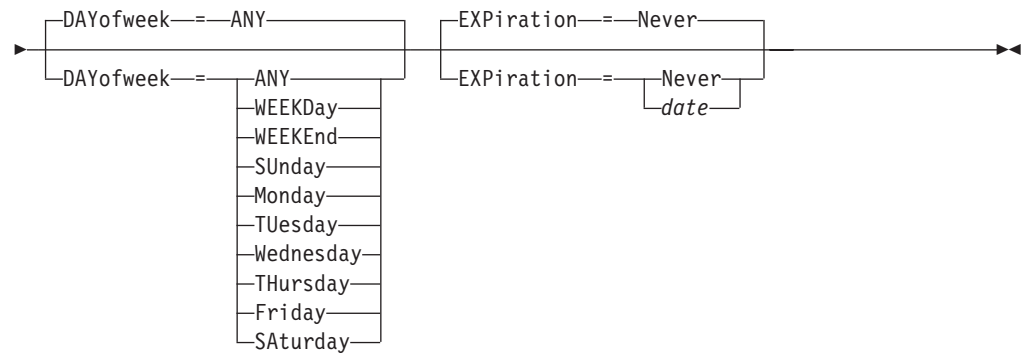
### Privilege class

To define an administrative command schedule, you must have system privilege.

### Syntax

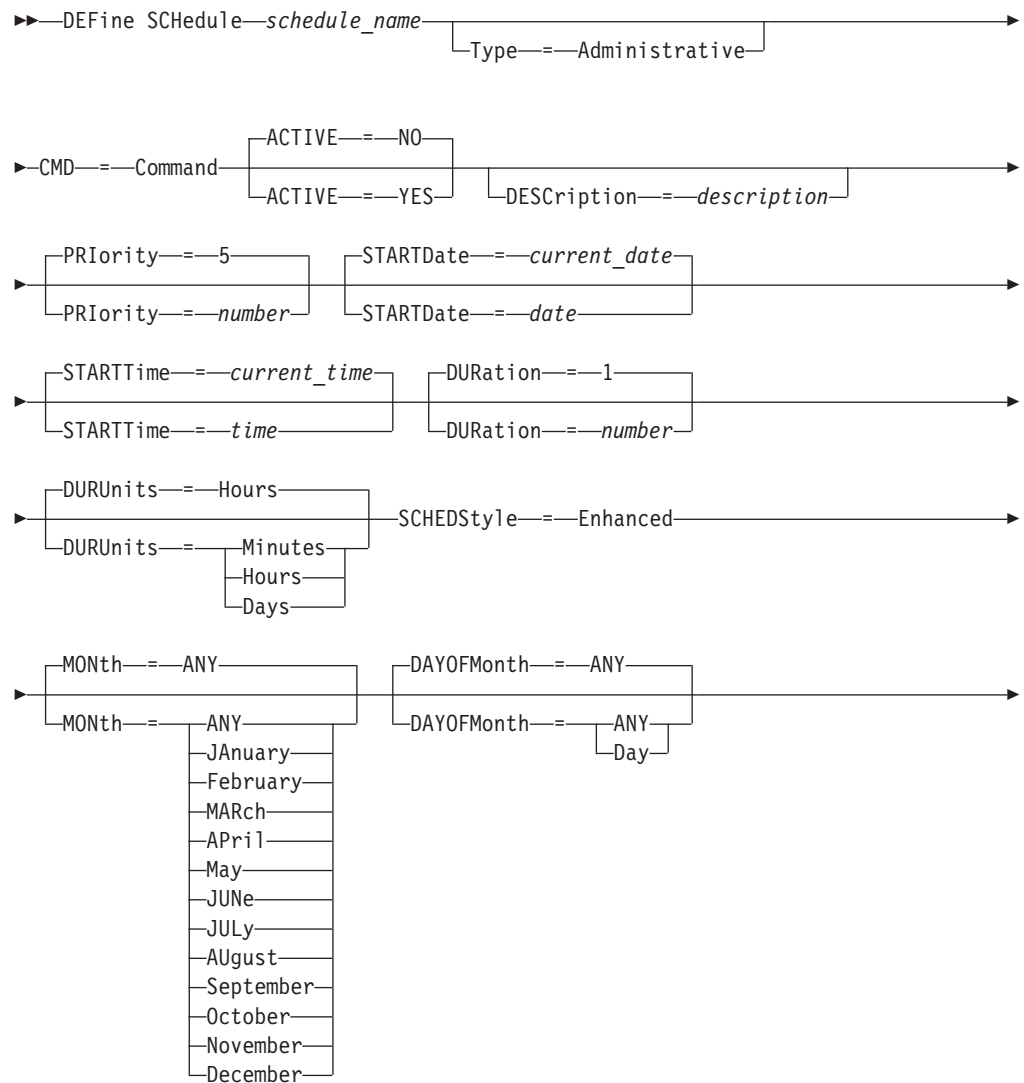
#### Classic administrative schedule



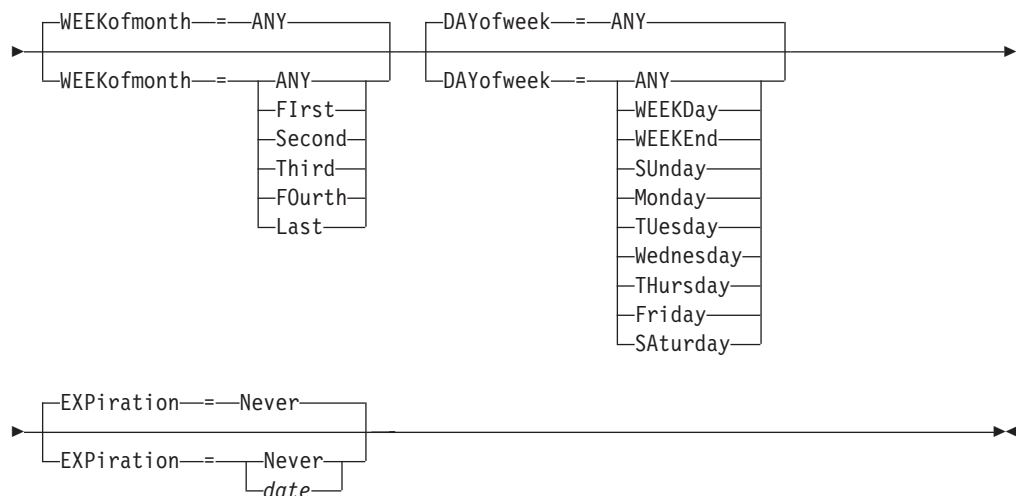


## Syntax

### Enhanced administrative schedule



## DEFINE SCHEDULE



### Parameters

#### *schedule\_name* (Required)

Specifies the name of the schedule to be defined. You can specify up to 30 characters for the name.

#### Type=Administrative

Specifies that a schedule for an administrative command is defined. This parameter is optional. An administrative command is assumed if the CMD parameter is specified.

#### CMD (Required)

Specifies the administrative command to schedule for processing. The maximum length of the command is 512 characters. Enclose the administrative command in quotation marks if it contains any blank characters.

**Note:** You cannot specify redirection characters with this parameter.

#### ACTIVE

Specifies whether Tivoli Storage Manager processes an administrative command schedule when the startup window occurs. This parameter is optional. The default is NO. The administrative command schedule must be set to the active state with the UPDATE SCHEDULE command so that Tivoli Storage Manager can process the schedule. Possible values are:

##### YES

Specifies that Tivoli Storage Manager processes an administrative command schedule when the startup window begins.

##### NO

Specifies that Tivoli Storage Manager does not process an administrative command schedule when the startup window begins.

#### DESCRIption

Specifies a description of the schedule. This parameter is optional. You can specify up to 255 characters for the description. Enclose the description in quotation marks if it contains any blank characters.

#### PRIority

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer from 1 to 10, with 1 being the highest priority and 10 being the lowest. The default is 5.



If two or more schedules have the same window start time, the value you specify determines when Tivoli Storage Manager processes the schedule. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

## STARTDate

Specifies the date for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current date. Use this parameter with the STARTTIME parameter to specify when the initial startup window of the schedule starts.

You can specify the date using one of the values below:

| Value                             | Description   | Example         |
|-----------------------------------|---|-----------------|
| <i>MM/DD/YYYY</i>                 | A specific date   | 09/15/1998      |
| <b>TODAY</b>                      | The current date  | TODAY           |
| <b>TODAY+days</b> or <i>+days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3. |

## STARTTime

Specifies the time for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current time. This parameter is used in conjunction with the STARTDATE parameter to specify when the initial startup window begins.

You can specify the time using one of the values below:

| Value                             | Description  | Example   |
|-----------------------------------|--|---|
| <i>HH:MM:SS</i>                   | A specific time                                    | 10:30:08  |
| <b>NOW</b>                        | The current time                                   | NOW   |
| <b>NOW+HH:MM</b> or <i>+HH:MM</i> | The current time plus hours and minutes specified  | NOW+02:00 or +02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW+02:00 or STARTTIME=+02:00, the beginning of the startup window is at 7:00. |
| <b>NOW-HH:MM</b> or <i>-HH:MM</i> | The current time minus hours and minutes specified | NOW-02:00 or -02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW-02:00 or STARTTIME=-02:00, the beginning of the startup window is at 3:00. |

## DURation

Specifies the number of units that define the length of the startup window of the scheduled operation. This parameter is optional. This value must be from 1 to 999. The default is 1.

Use this parameter 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 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**

Specifies the time units used to determine the duration of the window in which the schedule can start. This parameter is optional. The default is HOURS.

Use this parameter with the DURATION parameter to specify how long the startup window remains open to process 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 schedule may not necessarily complete processing within this window. If the schedule needs to be retried for any reason, the retry attempts must begin before the startup window elapses, or the operation does not restart.

The default value for the length of the startup window is 1 hour. 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 of the scheduled operation 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 specify PERUNITS=ONETIME. The INDEFINITE value is not allowed with enhanced schedules.

### **SCHEDStyle**

This parameter is optional. SCHEDSTYLE defines either the interval between times when a schedule should run, or the days on which it should run. The style can be either **classic** or **enhanced**. The default is the classic syntax.

For classic schedules, these parameters are allowed: PERIOD, PERUNITS, and DAYOFWEEK. Not allowed for classic schedules are: MONTH, DAYOFMONTH, and WEEKOFMONTH.

For enhanced schedules, these parameters are allowed: MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK. These parameters are not allowed: PERIOD and PERUNITS.

### **PERiod**

Specifies the length of time between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. You can specify an integer from 1 to 999. The default is 1.

Use this parameter 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 five days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default is 1 day.

This value is ignored if you specify PERUNITS=ONETIME.

## PERUnits

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. The default is DAYS.

Use this parameter 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 default is 1 day. Possible values are:

### Hours

Specifies that the time between startup windows is in hours.

### Days

Specifies that the time between startup windows is in days.

### Weeks

Specifies that the time between startup windows is in weeks.

### Months

Specifies that the time between startup windows is in months.

When you specify PERUNITS=MONTHS, the scheduled operation will be processed each month on the same date. For example, if the start date for the scheduled operation is 02/04/1998, the schedule will process on the 4th of every month thereafter. However, if the date is not valid for the next month, then the scheduled operation will be processed on the last valid date in the month. Thereafter, subsequent operations are based on this new date. For example, if the start date is 03/31/1998, the next month's operation will be scheduled for 04/30/1998. Thereafter, all subsequent operations will be on the 30th of the month until February. Because February has only 28 days, the operation will be scheduled for 02/28/1999. Subsequent operations will be processed on the 28th of the month.

### Years

Specifies that the time between startup windows for the schedule is in years.

When you specify PERUNITS=YEARS, the scheduled operation will be processed on the same month and date of each year. For example, if the start date for the scheduled operation is 02/29/2004, the next year's scheduled operation will be 02/28/2005 because February only has 28 days. Thereafter, subsequent operations will be scheduled for February 28th.

### Onetime

Specifies that the schedule processes once. This value overrides the value you specified for the PERIOD parameter.

## DAYofweek

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can specify different options for the DAYofweek parameter, depending on whether the schedule style has been defined as Classic or Enhanced:

### Classic Schedule

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can either specify one day of the week, or WEEKDAY, WEEKEND, or ANY. If the start date and start time fall on a day that does not correspond to a day you

## DEFINE SCHEDULE

specify, the start date and start time 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 processed when you would expect. The default is ANY.

### Enhanced Schedule

Specifies the days of the week on which to run the schedule. You can either specify multiple days separated by commas and no intervening blanks, or WEEKDAY, WEEKEND, or ANY. If you specify multiple days, the schedule will run on each of the specified days. If you specify WEEKDAY or WEEKEND, you must also specify either WEEKOFMONTH=FIRST or WEEKOFMONTH=LAST, and the schedule will run just once per month.

The default value is ANY, meaning the schedule will run every day of the week or on the day or days determined by other enhanced schedule parameters. DAYOFWEEK must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

Possible values for the DAYofweek parameter 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.

#### Saturday

Specifies that the startup window begins on Saturday.

### MONTH

Specifies the months of the year during which to run the schedule. This parameter is used only with enhanced schedules. Specify multiple values by using commas and no intervening blanks. The default value is ANY. This means the schedule will run during every month of the year.

### DAYOFMonth

Specifies the day of the month to run the schedule. This parameter is used only with enhanced schedules. You can either specify ANY or a number from

-31 through 31, excluding zero. Negative values are a day from the end of the month, counting backwards. For example, the last day of the month is -1, the next-to-the-last day of the month is -2, etc. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run on each of the specified days of the month. If multiple values resolve to the same day, the schedule will run only once that day.

The default value is ANY. This means the schedule will run on every day of the month or on the days determined by other enhanced schedule parameters. DAYOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFWEEK or WEEKOFMONTH parameters.

### WEEKofmonth

Specifies the week of the month in which to run the schedule. This parameter is used only with enhanced schedules. A week is considered any seven-day period which does not start on a particular day of the week. You can specify FIRST, SECOND, THIRD, FOURTH, LAST, or ANY. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run during each of the specified weeks of the month. If multiple values resolve to the same week, the schedule will run only once during that week.

The default value is ANY, meaning the schedule will run during every week of the month or on the day or days determined by other enhanced schedule parameters. WEEKOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

### EXPIRATION

Specifies the date after which this schedule is no longer used. This parameter is optional. The default is NEVER. Possible values are:

#### Never

Specifies that the schedule never expires.

#### *expiration\_date*

Specifies the date on which this schedule expires, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the date you specify.

### Example: Define a schedule to back up the primary storage pool every two days

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.

```
define schedule backup_archivepool type=administrative
cmd="backup stgpool archivepool recoverypool"
active=yes starttime=20:00 period=2
```

### Example: Define a schedule to back up the primary storage pool twice a month

Define a schedule named BACKUP\_ARCHIVEPOOL that backs up the primary storage pool ARCHIVEPOOL to the copy storage pool RECOVERYPOOL. Select an enhanced schedule and run on the first and fifteenth day of the month.

## DEFINE SCHEDULE

```
define schedule backup_archivepool type=administrative  
cmd="backup stgpool archivepool recoverypool"  
schedstyle=enhanced dayofmonth=1,15
```

## DEFINE SCRIPT (Define a Tivoli Storage Manager script)

Use this command to define a Tivoli Storage Manager script or to create a new Tivoli Storage Manager script using the contents from another script.

The first line for the script may be defined with this command. To add subsequent lines to the script, use the UPDATE SCRIPT command.

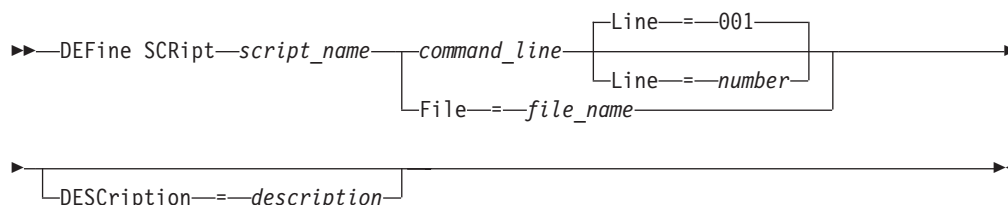
### Tip:

1. The Administration Center only supports ASCII characters for input. If you need to enter characters that are not ASCII, issue the DEFINE SCRIPT and UPDATE SCRIPT commands from the server console.
2. When routing commands inside scripts, enclose the server or server group in parentheses and omit the colon. Otherwise, if the syntax includes a colon, the command is not routed when the RUN command is issued. Instead, the command will only run on the server from which the RUN command is issued. For more information, see “Performing tasks concurrently on multiple servers” on page 16.

### Privilege class

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

### Syntax



### Parameters

#### *script\_name* (Required)

Specifies the name of the script to be defined. You can specify up to 30 characters for the name.

#### *command\_line*

Specifies the first command to be processed in a script. You must specify either this parameter (and, optionally, the LINE parameter) or the FILE parameter.

The command you specify can include substitution variables and can be continued across multiple lines if you specify a continuation character (-) as the last character in the command. Substitution variables are specified with a '\$' character, followed by a number that indicates the value of the parameter when the script is processed. You can specify up to 1200 characters for the command line. Enclose the command in quotation marks if it contains blanks.

Conditional logic flow statements can be used. These statements include IF, EXIT, and GOTO. For more information, see the *Administrator's Guide*. For return codes used with the IF statement, see Appendix A, “Return codes for use in IBM Tivoli Storage Manager scripts,” on page 1277.

## DEFINE SCRIPT

### Line

Specifies the line number for the command line. Because commands are specified in multiple lines, line numbers are used to determine the order for processing when the script is run. The first line, or line 001 is the default. This parameter is optional.

### File

Specifies the name of the file whose contents will be read into the script to be defined. The file must reside on the server running this command. If you specify the FILE parameter, you cannot specify a command line or line number.

You can create a script by querying another script and specifying the FORMAT=RAW and OUTPUTFILE parameters. The output from querying the script is directed to a file you specify with the OUTPUTFILE parameter. To create the new script, the contents of the script to be defined are read in from the file you specified with the OUTPUTFILE parameter.

### DEScription

Specifies a description for the script. You can specify up to 255 characters for the description. Enclose the description in quotation marks if it contains blank characters. This parameter is optional.

## Example: Write a script to display AIX clients

Define a script that will display all AIX clients.

```
define script qaixc "select node_name from nodes where platform_name='AIX'"  
  desc='Display aix clients'
```

## Example: Write and run a script to route a command to a server group

Define and run a script that will route the QUERY STGPOOL command to a server group named DEV\_GROUP.

```
define script qu_stg "(dev_group) query stgpool"  
run qu_stg
```

## Example: Create a script from an existing script

Define a script whose command lines are read in from a file that is named MY.SCRIPT and name the new script AGADM.

```
define script agadm file=my.script
```

## Related commands

*Table 102. Commands related to DEFINE SCRIPT*

| Command       | Description   |
|---------------|---|
| COPY SCRIPT   | Creates a copy of a script.                             |
| DELETE SCRIPT | Deletes the script or individual lines from the script. |
| QUERY SCRIPT  | Displays information about scripts.                     |
| RENAME SCRIPT | Renames a script to a new name.                         |
| RUN           | Runs a script.  |
| UPDATE SCRIPT | Changes or adds lines to a script.                      |



## DEFINE SERVER (Define a server for server-to-server communications)

Use this command to define a server.

Use this command to define a server for the following functions:

- Enterprise configuration
- Enterprise event logging
- Command routing
- Virtual volumes

The use of virtual volumes is not supported when the source server and the target server are the same Tivoli Storage Manager server.

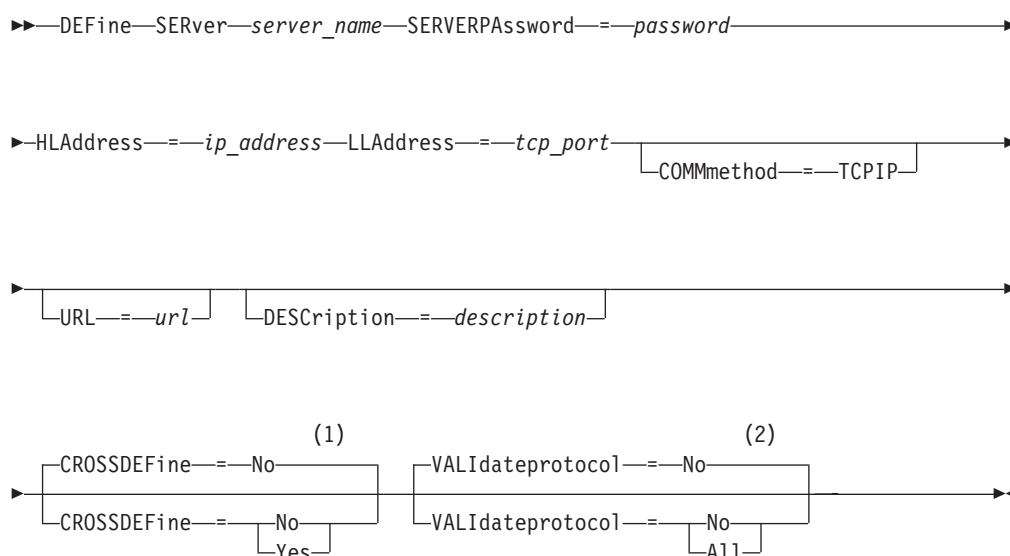
This command also is used to define an IBM Tivoli Storage Manager storage agent as if it were a server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

**For Enterprise Configuration, Enterprise Event Logging, Command Routing, and Storage Agent:**

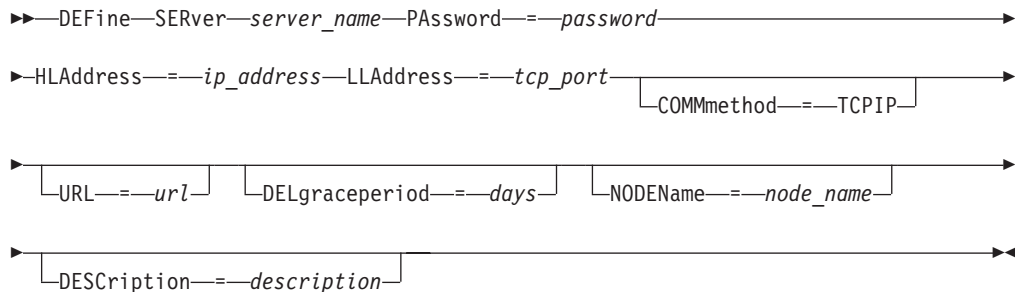


### Notes:

- 1 The CROSSDEFINE parameter does not apply to storage agent definitions.
- 2 The VALIDATEPROTOCOL parameter only applies to storage agent definitions.

## Syntax

For Virtual Volumes:



## Parameters

### *server\_name* (Required)

Specifies the name of the server. This name must be unique on the server. The maximum length of this name is 64 characters.

For command routing and server-to-server event logging functions, the server name you specify here should match the name that was set using the SET SERVERNAME command at the target server.

### **P**assword (Required)

Specifies the password used to sign on to the target server for virtual volumes. If you specify the NODENAME parameter, you must specify the PASSWORD parameter. If you specify the PASSWORD parameter but not the NODENAME parameter, the node name defaults to the server name specified with the SET SERVERNAME command.

### **S**ERVER**P**assword

Specifies the password of the server you are defining. This password must match the password set by the SET SERVERPASSWORD command. This parameter is required for enterprise configuration, command routing, and server-to-server event logging functions.

**Note:** Command routing uses the ID and password of the administrator issuing the command.

### **H**LAddress (Required)

Specifies the IP address (in dotted decimal format) of the server.

Do not use the loopback address as the value of this parameter. Virtual volumes are not supported when the source server and the target server are the same Tivoli Storage Manager server.

### **L**LAddress (Required)

Specifies the low-level address of the server. This address is usually the same as that in the TCPPORT server option of the target server.

### **C**OMMmethod

Specifies the communication method used to connect to the server. This parameter is optional.

### **U**RL

Specifies the URL address of this server. The parameter is optional.

## DELgraceperiod

Specifies a number of days that an object remains on the target server after it has been marked for deletion. Possible values are 0-9999. The default is 5. This parameter is optional.

## NODENAME

Specifies a node name to be used by the server to connect to the target server. This parameter is optional. If you specify the NODENAME parameter, you must also specify the PASSWORD parameter. If you specify the PASSWORD parameter but not the NODENAME parameter, the node name defaults to the server name specified with the SET SERVERNAME command.

## DESCRIPTION

Specifies a description of the server. The parameter is optional. The description can be up to 255 characters. Enclose the description in quotation marks if it contains blank characters.

## CROSSDEFINE

Specifies whether the server running this command will define itself to the server being specified by this command. This parameter is optional.

**Note:** This parameter does not apply to storage agent definitions.

If this parameter is included, you must also issue the SET SERVERNAME, SET SERVERPASSWORD, SET SERVERHLADDRESS, SET CROSSDEFINE, and SET SERVERLLADDRESS commands. The default is NO.

Possible values are:

### No

Cross definition is not to be performed.

### Yes

Cross definition is to be performed.

## VALIDATEprotocol

Specify whether a cyclic redundancy check should be performed to validate the data sent between the storage agent and Tivoli Storage Manager server. The parameter is optional. The default is NO. Possible values are:

### No

Specifies that data validation not be performed on any data sent between the storage agent and server.

### All

Specifies that data validation be performed on all client file data, client file metadata, and Tivoli Storage Manager server metadata that is sent between the storage agent and server. This mode impacts performance as additional overhead is required to calculate and compare CRC values between the storage agent and the server.

## Example: Define a target server

A target server has a high-level address of 9.116.2.67 and a low-level address of 1570. Define that target server to the source server, name it SERVER2, set the password to SECRET, and specify that objects remain on the target server for 7 days after they have been marked for deletion.

```
define server server2 password=secret
hladdress=9.115.3.45 lladdress=1570 delgraceperiod=7
```

### Example: Define a server to receive commands from other servers

Define a server to enable it to receive commands routed from other servers. Name the server WEST\_COMPLEX, set the password to CACTUS, and set the high-level address to 9.283.299, the low-level address to 1500, and the URL address to `http://west_complex:1580/`.

```
define server west_complex serverpassword=cactus
hladdress=9.283.299 lladdress=1500
url=http://west_complex:1580/
```

### Example: Cross define two servers

Use cross definition to define SERVER\_A and SERVER\_B.

1. On SERVER\_B, specify the server name, password, and high- and low-level addresses of SERVER\_B. Specify that cross defining is allowed.

```
set servername server_b
set serverpassword mylife
set serverhladdress 9.115.20.80
set serverlladdress 1860
set crossdefine on
```

2. On SERVER\_A, specify the server name, password, and high- and low-level addresses of SERVER\_A.

```
set servername server_a
set serverpassword yourlife
set serverhladdress 9.115.20.97
set serverlladdress 1500
```

3. On SERVER\_A, define SERVER\_B:

```
define server server_b hladdress=9.115.20.80 lladdress=1860
serverpassword=mylife crossdefine=yes
```

### Related commands

Table 103. Commands related to DEFINE SERVER

| Command           | Description  |
|-------------------|--|
| DEFINE DEVCLASS   | Defines a device class.  |
| DELETE DEVCLASS   | Deletes a device class name.   |
| DELETE FILESPACE  | Deletes data associated with client's file spaces.                                     |
| DELETE SERVER     | Deletes the definition of a server.  |
| QUERY NODE        | Displays partial or complete information about one or more clients.                    |
| QUERY SERVER      | Displays information about servers.  |
| RECONCILE VOLUMES | Reconciles source server virtual volume definitions and target server archive objects. |
| REGISTER NODE     | Defines a client to the server and sets options for that user.                         |
| REMOVE NODE       | Removes a client from the list of registered nodes for a specific policy domain.       |
| SET CROSSDEFINE   | Specifies whether to cross define servers.   |
| SET SERVERNAME    | Specifies the name by which the server is identified.                                  |

Table 103. Commands related to *DEFINE SERVER* (continued)

| Command             | Description   |
|---------------------|---|
| SET SERVERHLADDRESS | Specifies the high-level address of a server.         |
| SET SERVERLLADDRESS | Specifies the low-level address of a server.          |
| SET SERVERPASSWORD  | Specifies the server password.                        |
| UPDATE DEVCLASS     | Changes the attributes of a device class.             |
| UPDATE NODE         | Changes the attributes associated with a client node. |
| UPDATE SERVER       | Updates information about a server.                   |

## DEFINE SERVERGROUP (Define a server group)

Use this command to define a server group. A server group lets you route commands to multiple servers by specifying only the group name. After defining the server group, add servers to the group by using the DEFINE GRPMEMBER command.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—DEfIne SERVERGRoup—group_name—[DESCRiption==description]—◄◄
```

### Parameters

#### group\_name (Required)

Specifies the name of the server group. The maximum length of the name is 64 characters.

#### DESCRiption

Specifies a description of the server group. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

### Example: Define a server group

Define a server group named WEST\_COMPLEX.

```
define servergroup west_complex
```

### Related commands

Table 104. Commands related to DEFINE SERVERGROUP

| Command            | Description                                     |
|--------------------|---|
| COPY SERVERGROUP   | Creates a copy of a server group.               |
| DEFINE GRPMEMBER   | Defines a server as a member of a server group. |
| DELETE GRPMEMBER   | Deletes a server from a server group.           |
| DELETE SERVERGROUP | Deletes a server group.                         |
| MOVE GRPMEMBER     | Moves a server group member.                    |
| QUERY SERVERGROUP  | Displays information about server groups.       |
| RENAME SERVERGROUP | Renames a server group.                         |
| UPDATE SERVERGROUP | Updates a server group.                         |

## DEFINE SPACETRIGGER (Define the space trigger)

Use this command to define settings for triggers that determine when and how the server prepares additional space when predetermined thresholds have been exceeded in storage pools that use FILE and DISK device classes. Space triggers are not enabled for storage pools with a parameter RECLAMATIONTYPE=SNAPLOCK.

Tivoli Storage Manager allocates more space when space utilization reaches a specified value. After allocating more space, Tivoli Storage Manager either adds the space to the specified pool (random-access or sequential-access disk).

**Important:** Space trigger functions and storage pool space calculations take into account the space remaining in each directory. An inaccurate calculation could result in a failure to expand the space available in a storage pool. Failure to expand space in a storage pool is one of the conditions that can cause a trigger to become disabled.

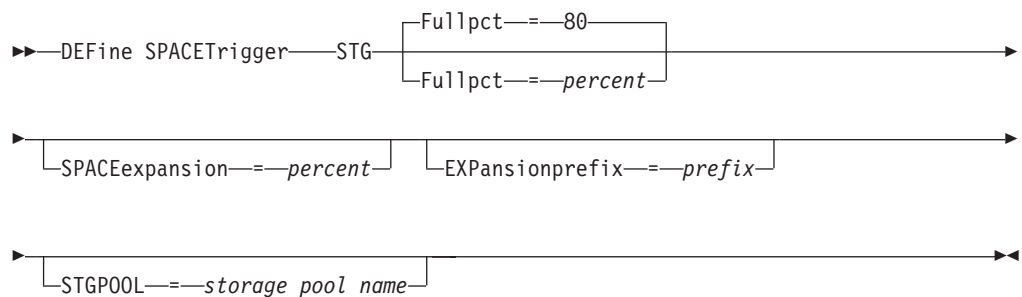
For example, if you specify multiple directories for a device class and the directories reside in the same file system, the server will calculate space by adding values representing the space remaining in each directory. These space calculations will be inaccurate. Rather than choosing a storage pool with sufficient space for an operation, the server could choose the wrong storage pool and run out of space prematurely.

To prevent possible problems and ensure an accurate calculation, you associate each directory with a separate file system. If a trigger becomes disabled because the space in a storage pool could not be expanded, you can re-enable the trigger by specifying the following command: update spacetrigger stg. No further changes are required to the space trigger.

### Privilege class

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

### Syntax



### Parameters

#### STG

Specifies a storage pool space trigger.

#### Fullpct

This parameter specifies the utilization percentage of the storage pool. This

## DEFINE SPACETRIGGER

parameter is optional. Specify an integer value from 0 to 99. The default is 80. A value of zero (0) disables the space trigger. When this value is exceeded, the space trigger creates new volumes. Exceeding the threshold may not cause new volumes to be created until the next space request is made.

You can determine storage pool utilization by issuing the QUERY STGPOOL command with FORMAT=DETAILED. The percentage of storage pool utilization is displayed in the field "Space Trigger Util." The calculation for this percentage does not include potential scratch volumes. The calculation for the percentage utilization used for migration and reclamation, however, does include potential scratch volumes.

### SPACEexpansion

For sequential-access FILE-type storage pools, this parameter is used in determining the number of additional volumes that are created in the storage pool. Volumes are created using the MAXCAPACITY value from the storage pool's device class. For random-access DISK storage pools, the space trigger creates a single volume using the EXPANSIONPREFIX.

### EXPansionprefix

For random-access DISK storage-pools, this parameter specifies the prefix that the server uses to create new storage pool files. This parameter is optional and applies only to random-access DISK device classes. The default prefix is the server installation path.

The prefix can include one or more directory separator characters, for example:  
`/opt/tivoli/tsm/server/bin/`

You can specify up to 250 characters. If you specify an invalid prefix, automatic expansion can fail.

This parameter is not valid for space triggers for sequential-access FILE storage pools. Prefixes are obtained from the directories specified with the associated device class.

### STGPOOL

Specifies the storage pool associated with this space trigger. This parameter is optional for storage pool space triggers. If you specify the STG parameter but not the STGPOOL parameter, one space trigger is created that applies to all random-access DISK and sequential-access FILE storage pools that do not have a specific space trigger.

This parameter does not apply to storage pools with the parameter RECLAMATIONTYPE=SNAPLOCK.

### Example: Define a space trigger to increase storage pool space 25 percent

Set up a storage pool space trigger for increasing the amount of space in a storage pool by 25 percent when it is filled to 80 percent utilization of existing volumes. Space will be created in the directories associated with the device class.

```
define spacetrigger stg spaceexpansion=25 stgpool=file
```

### Example: Define a space trigger to increase storage pool space 40 percent

Set up a space trigger for the WINPOOL1 storage pool to increase the amount of space in the storage pool by 40 percent when it is filled to 80 percent utilization of existing volumes.



```
define spacetrigger stg spaceexpansion=40 stgpool=winpool1
```

### Related commands

*Table 105. Commands related to DEFINE SPACETRIGGER*

| Command             | Description  |
|---------------------|--|
| DEFINE VOLUME       | Assigns a volume to be used for storage within a specified storage pool. |
| DELETE SPACETRIGGER | Deletes the storage pool space trigger.                                  |
| QUERY SPACETRIGGER  | Displays information about a storage pool space trigger.                 |
| UPDATE SPACETRIGGER | Changes attributes of storage pool space trigger.                        |

## DEFINE STGPOOL (Define a storage pool)

Use this command to define a primary storage pool, copy storage pool, or an active-data pool. A primary storage pool provides a destination for backup files, archive files, or files migrated from client nodes. A copy storage pool provides a destination for backup copies of files that are in primary storage pools. An active-data pool provides a destination for active versions of backup data that are in primary storage pools.

All volumes in a storage pool belong to the same device class. Random access storage pools use the DISK device type. After you define a random access storage pool, you must define volumes for the pool to create storage space.

Sequential access storage pools use device classes that you define for tape devices, optical devices, files on disk (FILE device type), and storage on another server (SERVER device type). To create storage space in a sequential access storage pool, you must allow scratch volumes for the pool when you define or update it, or define volumes for the pool after you define the pool. You can also do both.

The DEFINE STGPOOL command takes four forms:

- Defining a primary storage pool assigned to random access devices
- Defining a primary storage pool assigned to sequential access devices
- Defining a copy storage pool (always assigned to sequential access devices)
- Defining an active-data pool (always assigned to sequential access devices)

The syntax and parameters for each form are defined separately.

*Table 106. Commands related to DEFINE STGPOOL*

| Command             | Description   |
|---------------------|---|
| BACKUP DB           | Backs up the IBM Tivoli Storage Manager database to sequential access volumes.  |
| BACKUP STGPOOL      | Backs up a primary storage pool to a copy storage pool.                         |
| COPY ACTIVATEDATA   | Copies active backup data.  |
| DEFINE COLLOCGROUP  | Defines a collocation group.  |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                      |
| DEFINE DEVCLASS     | Defines a device class.   |
| DEFINE VOLUME       | Assigns a volume to be used for storage within a specified storage pool.        |
| DELETE COLLOCGROUP  | Deletes a collocation group.  |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                 |
| DELETE STGPOOL      | Deletes a storage pool from server storage.                                     |
| MOVE DATA           | Moves data from a specified storage pool volume to another storage pool volume. |
| MOVE MEDIA          | Moves storage pool volumes that are managed by an automated library.            |
| QUERY COLLOCGROUP   | Displays information about collocation groups.                                  |
| QUERY DEVCLASS      | Displays information about device classes.                                      |

Table 106. Commands related to *DEFINE STGPOOL* (continued)

| Command            | Description   |
|--------------------|---|
| QUERY NODEDATA     | Displays information about the location and size of data for a client node.                   |
| QUERY SHREDSTATUS  | Displays information about data waiting to be shredded.                                       |
| QUERY STGPOOL      | Displays information about storage pools.   |
| RENAME STGPOOL     | Renames a storage pool.   |
| RESTORE STGPOOL    | Restores files to a primary storage pool from copy storage pools.                             |
| RESTORE VOLUME     | Restores files stored on specified volumes in a primary storage pool from copy storage pools. |
| SET DRMPRIMSTGPOOL | Specifies that primary storage pools are managed by DRM.                                      |
| SHRED DATA         | Manually starts the process of shredding deleted data.  |
| UPDATE COLLOCGROUP | Updates the description of a collocation group.   |
| UPDATE STGPOOL     | Changes the attributes of a storage pool.   |

## DEFINE STGPOOL

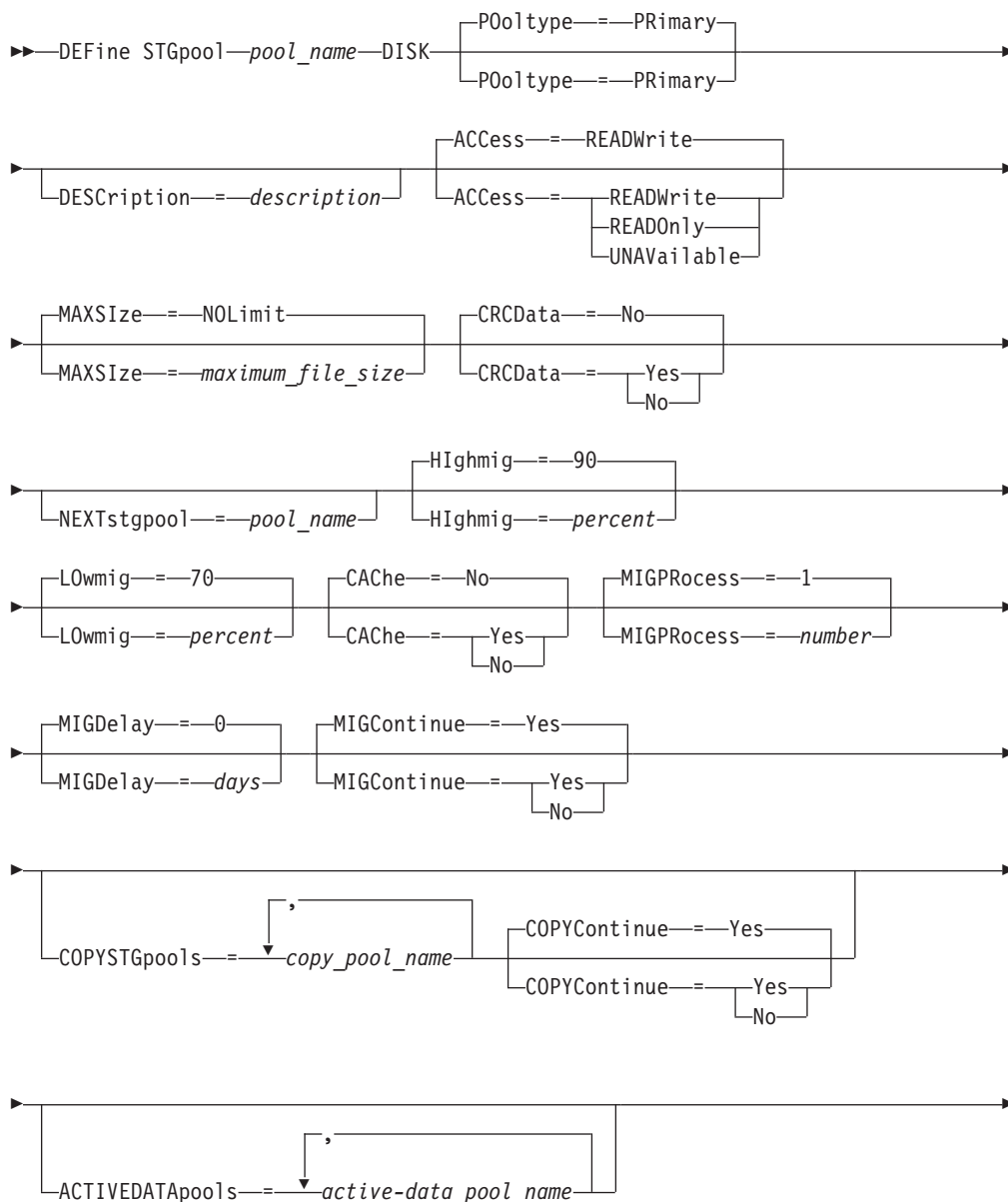
### DEFINE STGPOOL (Define a primary storage pool assigned to random access devices)

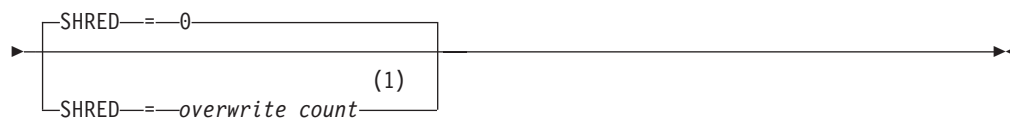
Use this command to define a primary storage pool assigned to random access devices.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax





**Notes:**

- 1 This parameter is not available for Centera or SnapLock storage pools.

**Parameters**

***pool\_name* (Required)**

Specifies the name of the storage pool to be defined. The name must be unique, and the maximum length is 30 characters.

**DISK (Required)**

Specifies that you want to define a storage pool to the DISK device class (the DISK device class is predefined during installation).

**Pooltype=Primary**

Specifies that you want to define a primary storage pool. This parameter is optional. The default value is PRIMARY.

**DEScRiption**

Specifies a description of the storage pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

**ACcEss**

Specifies how client nodes and server processes (such as 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 client nodes and server processes can read and write to files stored on volumes in the storage pool.

**READOnly**

Specifies that client nodes can only read files from the volumes in the storage pool.

Server processes can move files within the volumes in the storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside 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 server processes attempt to write files to the storage pool.

**UNAVailable**

Specifies that client nodes cannot access files stored on volumes in the storage pool.

Server processes can move files within the volumes in the storage pool and can also move or copy files from this storage pool to another storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside the 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 server processes attempt to write files to the storage pool.

## DEFINE STGPOOL

### MAXSize

Specifies the maximum size for a physical file that the server can store in the storage pool. This parameter is optional. The default value is NOLIMIT. Possible values are:

#### NOLimit

Specifies that there is no maximum size limit for physical files stored in the storage pool.

#### *maximum\_file\_size*

Limits the maximum physical file size. Specify an integer from 1 to 999999, followed by a scale factor. For example, MAXSIZE=5G specifies that the maximum file size for this storage pool is 5 GB. Scale factors are:

| Scale factor | Meaning |
|--------------|---------|
|--------------|---------|

|   |          |
|---|----------|
| K | kilobyte |
| M | megabyte |
| G | gigabyte |
| T | terabyte |

If a file exceeds the maximum size and no pool is specified as the next storage pool in the hierarchy, the server does not store the file. If a file exceeds the maximum size and a pool is specified as the next storage pool, the server stores the file in the next storage pool that can accept the file size. If you specify the next storage pool parameter, at least one storage pool in your hierarchy should have no limit on the maximum size of a file. By having no limit on the size for at least one pool, you ensure that no matter what its size, the server can store the file.

For logical files that are part of an aggregate, the server considers the size of the aggregate to be the file size. Therefore, the server does not store logical files that are smaller than the maximum size limit if the files are part of an aggregate that is larger than the maximum size limit.

### CRCData

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

#### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

#### No

Specifies that data is stored without CRC information.

### NEXTstgpool

Specifies a primary storage pool to which files are migrated. This parameter is optional.

If you do not specify a next storage pool, the server cannot migrate files from this storage pool and cannot store files that exceed the maximum size for this storage pool 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**

Specifies that the server starts migration for this storage pool when the amount of data in the pool reaches this percentage of the pool's estimated capacity. This parameter is optional. You can specify an integer from 0 to 100. The default value is 90.

When the storage pool exceeds the high migration threshold, the server can start migration of files 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.

### **LOWmig**

Specifies that the server stops migration for this storage pool when the amount of data in the pool reaches this percentage of the pool's estimated capacity. This parameter is optional. You can specify an integer from 0 to 99. The default value is 70.

When the storage pool reaches the low migration threshold, the server does not start 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**

Specifies whether the migration process leaves a cached copy of a file in this storage pool after migrating the file to the next storage pool. This parameter is optional. The default value is NO. Possible values are:

#### **Yes**

Specifies that caching is enabled.

#### **No**

Specifies that caching is disabled.

Using cache may improve the retrievability of files, but may affect the performance of other processes. See the *Administrator's Guide* for details.

### **MIGProcess**

Specifies the number of processes that the server uses for migrating files from this storage pool. This parameter is optional. You can specify an integer from 1 to 999. The default value is 1.

During migration, the server runs this number of processes in parallel to provide the potential for improved migration rates.

**Note:** The number of migration processes is dependent upon the setting of the MIGPROCESS parameter and the number of nodes or the number of collocation groups with data in the migrating storage pool. For example, if the MIGPROCESS parameter is equal to six, but there are only two nodes with data on the storage pool, migration processing only consists of two processes, not six.

### **MIGDelay**

Specifies the minimum number of days a file must remain in a storage pool before it becomes eligible for migration. To calculate a value to compare to the specified MIGDELAY value, the server counts the number of days that the file has been in the storage pool and the number of days, if any, since the file was

retrieved by a client. The lesser of the two values is compared to the specified MIGDELAY value. For example, if all the following conditions are true, a file is not migrated:

- A file has been in a storage pool for five days.
- The file was accessed by a client within the past three days.
- The value specified for the MIGDELAY parameter is four days.

This parameter is optional. You can specify an integer from 0 to 9999. The default is 0, which means that you do not want to delay migration.

If you want the server to count the number of days based only on when a file was stored and not when it was retrieved, use the NORETRIEVEDATE server option.

### **MIGContinue**

Specifies whether you allow the server to migrate files that do not satisfy the migration delay time. This parameter is optional. The default is YES.

Because you can require that files remain in the storage pool for a minimum number of days, the server may migrate all eligible files to the next storage pool yet not meet the low migration threshold. This parameter allows you to specify whether the server is allowed to continue the migration process by migrating files that do not satisfy the migration delay time.

Possible values are:

#### **Yes**

Specifies that, when necessary to meet the low migration threshold, the server continues to migrate files that do not satisfy the migration delay time.

If you allow more than one migration process for the storage pool, some files that do not satisfy the migration delay time may be migrated unnecessarily. As one process migrates files that satisfy the migration delay time, a second process could begin migrating files that do not satisfy the migration delay time to meet the low migration threshold. The first process that is still migrating files that satisfy the migration delay time might have, by itself, caused the low migration threshold to be met.

#### **No**

Specifies that the server stops migration when no eligible files remain to be migrated, even before reaching the low migration threshold. The server does not migrate files unless the files satisfy the migration delay time.

### **COPYSTGpools**

Specifies the names of copy storage pools where the server simultaneously writes data. The COPYSTGPools parameter is optional. Beginning with Tivoli Storage Manager Version 5.3, you can specify a maximum of three copy pool names separated by commas. (In versions earlier than Version 5.3, the maximum number was ten.) Spaces between the names of the copy pools are not permitted. When specifying a value for the COPYSTGPools parameter, you can also specify a value for the COPYCONTINUE parameter.

The combined total number of storage pools specified in the COPYSTGPools and ACTIVEDATAPools parameters cannot exceed three.

The simultaneous write function changed in Tivoli Storage Manager Version 5.3. In Version 5.3 and later, when a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of copy storage pools and the COPYCONTINUE value from the primary storage pool. The primary storage pool is specified by the copy group of the



management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to copy storage pools for the following operations:

- Backup and archive operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API
- Migration operations by Tivoli Storage Manager for Space Management clients
- Import operations that involve copying exported file data from external media to a primary storage pool associated with a copy storage pool list

**Note:** Simultaneous write is not supported for the following store operations:

- When the operation is using LAN-free data movement. Simultaneous write takes precedence over LAN-free operations, causing the operations to go over the LAN. However, the simultaneous write configuration is honored.
- NAS backup operations. If the primary storage pool specified in the DESTINATION or TOCDESTINATION in the copy group of the management class has copy storage pools defined, the copy storage pools are ignored and the data is stored into the primary storage pool only.

**Attention:** The function provided by the COPYSTGPOLLS parameter is not intended to replace the BACKUP STGPOOL command. If you use the COPYSTGPOLLS parameter, continue to use the BACKUP STGPOOL command to ensure that the copy storage pools are complete copies of the primary storage pool. There are cases when a copy might not be created. For more information, see the COPYCONTINUE parameter description.

### COPYContinue

Specifies how the server should react to a copy storage pool write failure for any of the copy storage pools listed in the COPYSTGPOLLS parameter. This parameter is optional. The default value is YES. When specifying the COPYCONTINUE parameter, you must also specify the COPYSTGPOLLS parameter.

This parameter has no effect on active-data pools. If a write error occurs while writing to an active-data pool, the server will stop writing to the failing active-data pool for the remainder of the session.

Possible values are:

#### Yes

If the COPYCONTINUE option is set to YES, the server will stop writing to the failing copy pools for the remainder of the session, but continue storing files into the primary pool and any remaining copy pools. The copy storage pool list is active only for the life of the client session and applies to all the primary storage pools in a particular storage pool hierarchy.

For additional information about the COPYCONTINUE parameter, refer to the information about simultaneous write in the *Administrator's Guide*.

#### No

If the COPYCONTINUE option is set to NO, the server will fail the current transaction and discontinue the store operation.

### ACTIVEDATApools

Specifies the names of active-data pools where the server simultaneously

## DEFINE STGPOOL

writes data during a client backup operation. The `ACTIVEDATAPOOLS` parameter is optional. Spaces between the names of the active-data pools are not permitted.

The combined total number of storage pools specified in the `COPYSGTPOOLS` and `ACTIVEDATAPOOLS` parameters can not exceed three.

When a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of active-data pools from the destination storage pool specified in the copy group. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API.

### Restrictions:

1. This parameter is available only to primary storage pools that use `NATIVE` or `NONBLOCK` data format. This parameter is not available for storage pools that use the following data formats:
  - `NETAPPDUMP`
  - `CELERRADUMP`
  - `NDMPDUMP`
2. Simultaneous write to active-data pools is not supported when the operation is using LAN-free data movement. Simultaneous write takes precedence over LAN-free operations, causing the operations to go over the LAN. However, the simultaneous write configuration is honored.
3. Simultaneous write is not supported when a NAS backup operation is writing a TOC file. If the primary storage pool specified in the `TOCDESTINATION` in the copy group of the management class has active-data pools defined, the active-data pools are ignored and the data is stored into the primary storage pool only.
4. You cannot use the simultaneous write function with Centera storage devices.
5. Data being imported will not be stored in active-data pools. After an import operation, use the `COPY ACTIVEDATA` command to store the imported data in an active-data pool.

**Attention:** The function provided by the `ACTIVEDATAPOOLS` parameter is not intended to replace the `COPY ACTIVEDATA` command. If you use the `ACTIVEDATAPOOLS` parameter, use the `COPY ACTIVEDATA` command to ensure that the active-data pools contain all active data of the primary storage pool.

### SHRED

Specifies whether data will be physically overwritten when it is deleted. This parameter is optional. You can specify an integer from 0 to 10. The default value is 0.

If you specify a value of 0, the Tivoli Storage Manager server will delete the data from the database. However, the storage used to contain the data will not be overwritten, and the data will still exist in storage until that storage is reused for other data. It might be possible to discover and reconstruct the data after it has been deleted.

If you specify a value greater than 0, the Tivoli Storage Manager server will delete the data both logically and physically. The server will overwrite the storage used to contain the data the specified number of times. This increases the difficulty of discovering and reconstructing the data after it has been deleted.

To ensure that all copies of the data are shredded, specify a SHRED value greater than 0 for the storage pool specified in the NEXTSTGPOOL parameter, and do not specify either the COPYSTGPools or ACTIVEDATAPOOLS. Specifying relatively high values for the overwrite count will generally improve the level of security, but could affect performance adversely.

Overwriting of deleted data is performed asynchronously after the delete operation is complete. Therefore, the space occupied by the deleted data will remain occupied for some period of time and will not be available as free space for new data.

A SHRED value greater than zero cannot be used if the value of the CACHE parameter is YES.

**Important:** After an export operation has finished identifying files for export, any changes to the storage pool SHRED value is ignored. An export operation that is suspended retains the original SHRED value throughout the operation. You might want to consider cancelling your export operation if changes to the storage pool SHRED value jeopardize the operation. You can reissue the export command after any needed cleanup.

### Example: Define a primary storage pool for a DISK device class

Define a primary storage pool, POOL1, to use the DISK device class, with caching enabled. Limit the maximum file size to 5 MB. Store any files larger than 5 MB in subordinate storage pools beginning with the PROG2 storage pool. Set the high migration threshold to 70 percent, and the low migration threshold to 30 percent.

```
define stgpool pool1 disk
description="main disk storage pool" maxsize=5m
highmig=70 lowmig=30 cache=yes
nextstgpool=prog2
```

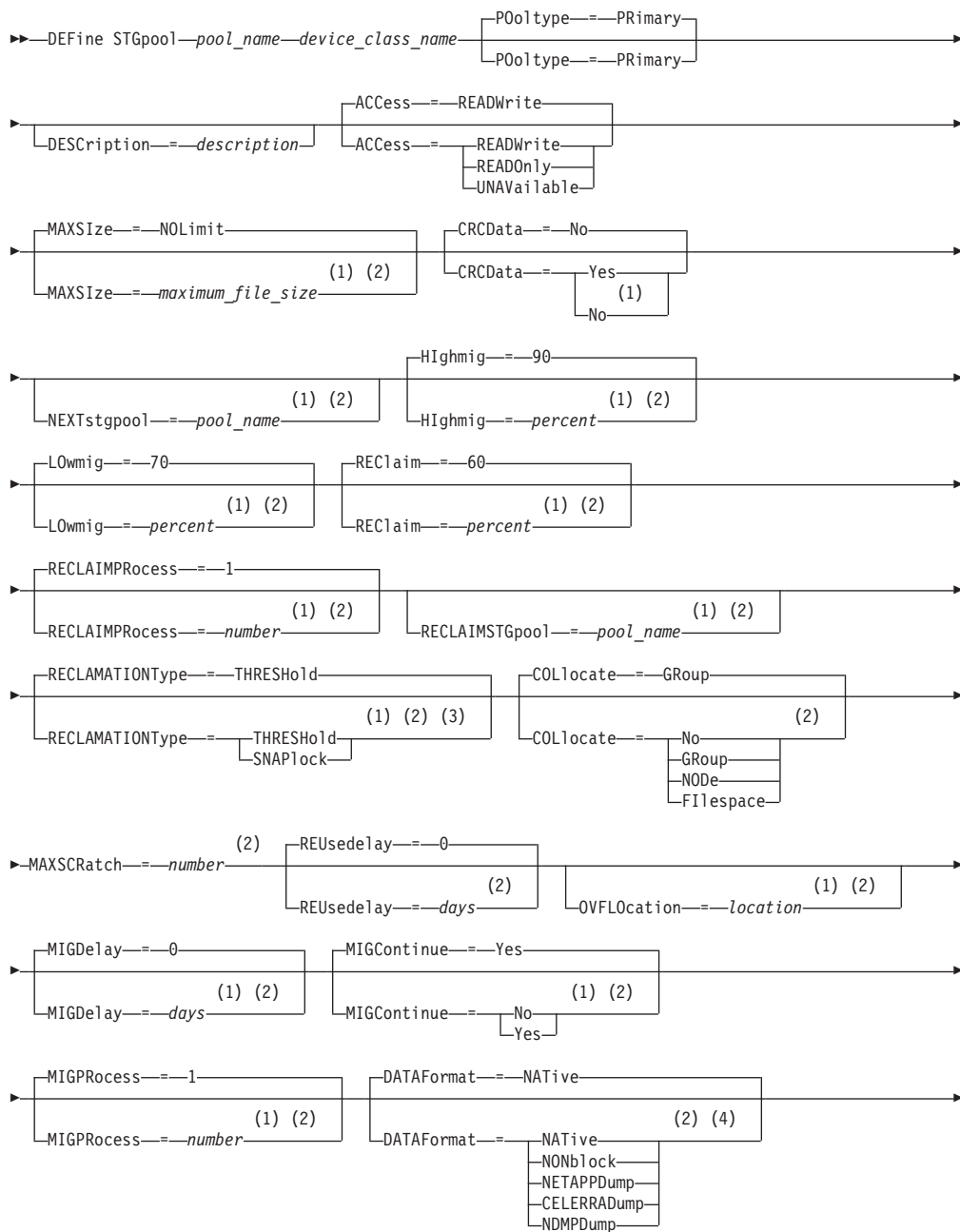
## **DEFINE STGPOOL (Define a primary storage pool assigned to sequential access devices)**

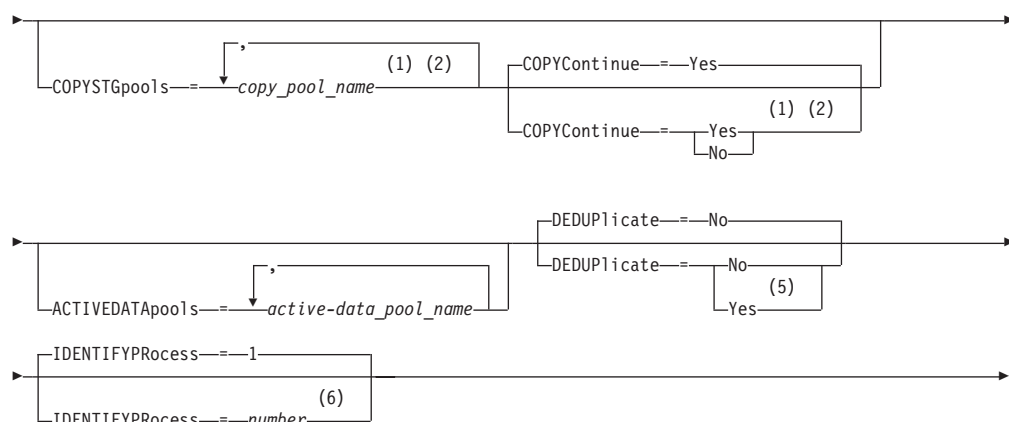
Use this command to define a primary storage pool assigned to sequential access devices.

### **Privilege class**

To issue this command, you must have system privilege.

### **Syntax**





## Notes:

- 1 This parameter is not available for storage pools that use the data formats NETAPPDUMP, CELERRADUMP, or NDMPDUMP.
- 2 This parameter is not available or is ignored for Centera storage pools.
- 3 The RECLAMATIONTYPE=SNAPLOCK setting is valid only for storage pools defined to servers that are enabled for System Storage Archive Manager. The storage pool must be assigned to a FILE device class, and the directories specified in the device class must be NetApp SnapLock volumes.
- 4 The values NETAPPDUMP, CELERRADUMP, and NDMPDUMP are not valid for storage pools defined with a FILE-type device class.
- 5 This parameter is valid only for storage pools that are defined with a FILE-type device class.
- 6 This parameter is available only when the value of the DEDUPLICATE parameter is YES.

## Parameters

### *pool\_name* (Required)

Specifies the name of the storage pool to be defined. The name must be unique, and the maximum length is 30 characters.

### *device\_class\_name* (Required)

Specifies the name of the device class to which this storage pool is assigned. You can specify any device class except for the DISK device class.

### **POoltype=PRimary**

Specifies that you want to define a primary storage pool. This parameter is optional. The default value is PRIMARY.

### **DESCription**

Specifies a description of the storage pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

### **ACCess**

Specifies how client nodes and server processes (such as migration and reclamation) can access files in the storage pool. This parameter is optional. The default value is READWRITE. Possible values are:

## DEFINE STGPOOL

### READWrite

Specifies that client nodes and server processes can read and write to files stored on volumes in the storage pool.

### READOnly

Specifies that client nodes can only read files from the volumes in the storage pool.

Server processes can move files within the volumes in the storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside 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 server processes attempt to write files to the storage pool.

### UNAVailable

Specifies that client nodes cannot access files stored on volumes in the storage pool.

Server processes can move files within the volumes in the storage pool and can also move or copy files from this storage pool to another storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside the 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 server processes attempt to write files to the storage pool.

### MAXSize

Specifies the maximum size for a physical file that the server can store in the storage pool. This parameter is optional. The default value is NOLIMIT. Possible values are:

#### NOLimit

Specifies that there is no maximum size limit for physical files stored in the storage pool.

#### *maximum\_file\_size*

Limits the maximum physical file size. Specify an integer from 1 to 999999, followed by a scale factor. For example, MAXSIZE=5G specifies that the maximum file size for this storage pool is 5 gigabytes. Scale factors are:

| Scale factor | Meaning |
|--------------|---------|
|--------------|---------|

|   |          |
|---|----------|
| K | kilobyte |
| M | megabyte |
| G | gigabyte |
| T | terabyte |

If a file exceeds the maximum size and no pool is specified as the next storage pool in the hierarchy, the server does not store the file. If a file exceeds the maximum size and a pool is specified as the next storage pool, the server stores the file in the next storage pool that can accept the file size. If you specify the next storage pool parameter, at least one storage pool in your hierarchy should have no limit on the maximum size of a file. By having no limit on the size for at least one pool, you ensure that no matter what its size, the server can store the file.

For logical files that are part of an aggregate, the server considers the size of the aggregate to be the file size. Therefore, the server does not store logical files that are smaller than the maximum size limit if the files are part of an aggregate that is larger than the maximum size limit.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

## CRCData

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is only valid for NATIVE data format storage pools. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

### No

Specifies that data is stored without CRC information.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

## NEXTstgpool

Specifies a primary storage pool to which files are migrated. You cannot migrate data from a sequential access storage pool to a random access storage pool. This parameter is optional.

If this storage pool does not have a next storage pool, the server cannot migrate files from this storage pool and cannot store files that exceed the maximum size for this storage pool in another storage pool.

When there is insufficient space available in the current storage pool, the NEXTSTGPOOL parameter for sequential access storage pools does not allow data to be stored into the next pool. In this case the server issues a message and the transaction fails.

For next storage pools with a device type of FILE, the server performs a preliminary check to determine whether sufficient space is available. If space is not available, the server skips to the next storage pool in the hierarchy. If space is available, the server attempts to store data in that pool. However, it is possible that the storage operation could fail because, at the time the actual storage operation is attempted, the space is no longer available.

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.

## DEFINE STGPOOL

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### Highmig

Specifies that the server starts migration when storage pool utilization reaches this percentage. For sequential-access disk (FILE) storage pools, utilization is the ratio of data in a storage pool to the pool's total estimated data capacity, including the capacity of all scratch volumes specified for the pool. For storage pools that use tape or optical media, utilization is the ratio of volumes that contain data to the total number of volumes in the storage pool. The total number of volumes includes the maximum number of scratch volumes. This parameter is optional. You can specify an integer from 0 to 100. The default value is 90.

When the storage pool exceeds the high migration threshold, the server can start migration of files by volume to the next storage pool defined for the pool. You can set the high migration threshold to 100 to prevent migration for the storage pool.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### LOwmig

Specifies that the server stops migration when storage pool utilization is at or below this percentage. For sequential-access disk (FILE) storage pools, utilization is the ratio of data in a storage pool to the pool's total estimated data capacity, including the capacity of all scratch volumes specified for the pool. For storage pools that use tape or optical media, utilization is the ratio of volumes that contain data to the total number of volumes in the storage pool. The total number of volumes includes the maximum number of scratch volumes. This parameter is optional. You can specify an integer from 0 to 99. The default value is 70.

When the storage pool reaches the low migration threshold, the server does not start migration of files from another volume. You can set the low migration threshold to 0 to permit migration to empty the storage pool.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### REClaim

Specifies when the server reclaims a volume, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space on volumes usable again by moving any remaining unexpired 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 to 100. The default value is 60, except for storage pools that use WORM devices.



For storage pools that use WORM devices, the default value is 100 to prevent reclamation from occurring. This is the default because a WORM volume is not reusable. If necessary, you can lower the value to allow the server to consolidate data onto fewer volumes. Volumes emptied by reclamation can be checked out of the library, freeing slots for new volumes.

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.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### RECLAIMPROCESS

Specifies the number of parallel processes to use for reclaiming the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999. The default value is 1.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the reclamation and the number of logical and physical drives that can be dedicated to the operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the reclamation.

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that you want to specify four processes for each of the storage pools. The storage pools have the same device class. Assuming that the RECLAIMSTGPOOL parameter is not specified or that the reclaim storage pool has the same device class as the storage pool being reclaimed, each process requires two mount points and, if the device type is not FILE, two drives. (One of the drives is for the input volume, and the other drive is for the output volume.) To run eight reclamation processes simultaneously, you need a total of at least 16 mount points and 16 drives. The device class for the storage pools must have a mount limit of at least 16.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the reclamation processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP

## DEFINE STGPOOL

- NDMPDUMP

### RECLAIMSTGpool

Specifies another primary storage pool as a target for reclaimed data from this storage pool. This parameter is optional. When the server reclaims volumes for the storage pool, the server moves unexpired data from the volumes being reclaimed to the storage pool named with this parameter.

A reclaim storage pool is most useful for a storage pool that has only one drive in its library. When you specify this parameter, the server moves all data from reclaimed volumes to the reclaim storage pool regardless of the number of drives in the library.

To move data from the reclaim storage pool back to the original storage pool, use the storage pool hierarchy. Specify the original storage pool as the next storage pool for the reclaim storage pool.

#### Restriction:

- This parameter is not available for storage pools that use the following data formats:
- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### RECLAMATIONType

Specifies the method by which volumes are reclaimed and managed. This parameter is optional. The default value is THRESHOLD. Possible values are the following:

#### THRESHold

Specifies that volumes belonging to this storage pool are reclaimed based on the threshold value in the RECLAIM attribute for this storage pool.

#### SNAPlock

Specifies that FILE volumes belonging to this storage pool will be managed for retention using NetApp Data ONTAP software and NetApp SnapLock volumes. This parameter is only valid for storage pools being defined to a server that has data retention protection enabled and that is assigned to a FILE device class. Volumes in this storage pool are not reclaimed based on threshold; the RECLAIM value for the storage pool is ignored.

All volumes in this storage pool are created as FILE volumes. A retention date, derived from the retention attributes in the archive copy group for the storage pool, is set in the metadata for the FILE volume using the SnapLock feature of the NetApp Data ONTAP operating system. Until the retention date has expired, the FILE volume and any data on it cannot be deleted from the physical SnapLock volume on which it is stored.

The RECLAMATIONTYPE parameter for all storage pools being defined must be the same when defined to the same device class name. The DEFINE command will fail if the RECLAMATIONTYPE parameter specified is different than what is currently defined for storage pools already defined to the device class name.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP

- NDMPDUMP

### **COLlocate**

Specifies whether the server attempts to keep data belonging to a single client node, group of client nodes, or client file space stored on as few volumes as possible. This parameter is optional. The default value is GROUP.

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. For details, see the *Administrator's Guide*.

Possible values are:

#### **No**

Specifies that collocation is disabled.

#### **GGroup**

Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible. If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

If you specify COLLOCATE=GROUP but do not define any collocation groups or if you specify COLLOCATE=GROUP but do not add nodes to a collocation group, data is collocated by node. Be sure to consider tape usage when organizing client nodes into collocation groups. For example, if a tape-based storage pool consists of data from grouped and ungrouped nodes and you specify COLLOCATE=GROUP, the server performs the following actions:

- Collocates by group the data for grouped nodes only. Whenever possible, the server collocates data belonging to a group of nodes on a single tape or on as few tapes as possible. Data for a single node can also be spread across several tapes associated with a group.
- Collocates by node the data for ungrouped nodes. Whenever possible, the server stores the data for a single node on a single tape. All available tapes that already have data for the node are used before available space on any other tape is used.

#### **NODE**

Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible. If the node has multiple file spaces, the server does not attempt to collocate those file spaces. For backward compatibility, COLLOCATE=YES is still accepted by the server to specify collocation at the client node level.

If a storage pool contains data for a node that is a member of a collocation group and you specify COLLOCATE=NODE, the data will be collocated by node not by group.

#### **Filespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

### **MAXSCRatch (Required)**

Specifies the maximum number of scratch volumes that the server can request

## DEFINE STGPOOL

for this storage pool. You can specify an integer from 0 to 100000000. By allowing the server to request scratch volumes, you avoid having to define each volume to be used.

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.

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 the server and returned to the file system.

**Tip:** For server-to-server operations that utilize virtual volumes and that store a small amount of data, consider specifying a value for the MAXSCRATCH parameter that is higher than the value you typically specify for write operations to other types of volumes. After a write operation to a virtual volume, Tivoli Storage Manager marks the volume as FULL, even if the value of the MAXCAPACITY parameter on the device-class definition has not been reached. The Tivoli Storage Manager server does not keep virtual volumes in FILLING status and does not append to them. If the value of the MAXSCRATCH parameter is too low, server-to-server operations can fail.

### REUsedelay

Specifies the number of days that must elapse after all files are 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 to 9999. The default value is 0, which means that a volume can be rewritten or returned to the scratch pool as soon as all the files are deleted from the volume.

**Important:** Use this parameter to help ensure that when you restore the database to an earlier level, database references to files in the storage pool are still valid. You must set this parameter to a value greater than the number of days you plan to retain the oldest database backup. The number of days specified for this parameter should be the same as the number specified for the SET DRMDBBACKUPEXPIREDAYS command. For more information, see the *Administrator's Guide*.

### OVFLocation

Specifies the overflow location for the storage pool. The server assigns this location name to a volume that is ejected from the library by the command. This parameter is optional. The location name can be a maximum length of 255 characters. Enclose the location name in quotation marks if the location name contains any blank characters.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### MIGDelay

Specifies the minimum number of days a file must remain in a storage pool before it becomes eligible for migration. All files on a volume must be eligible for migration before the server selects the volume for migration. To calculate a value to compare to the specified MIGDELAY, the server counts the number of days that the file has been in the storage pool.

This parameter is optional. You can specify an integer from 0 to 9999. The default is 0, which means that you do not want to delay migration. If you want the server to count the number of days based only on when a file was stored and not when it was retrieved, use the NORETRIEVEDATE server option.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

## MIGContinue

Specifies whether you allow the server to migrate files that do not satisfy the migration delay time. This parameter is optional. The default is YES.

Because you can require that files remain in the storage pool for a minimum number of days, the server may migrate all eligible files to the next storage pool yet not meet the low migration threshold. This parameter allows you to specify whether the server is allowed to continue the migration process by migrating files that do not satisfy the migration delay time.

Possible values are:

### Yes

Specifies that, when necessary to meet the low migration threshold, the server continues to migrate files that do not satisfy the migration delay time.

If you allow more than one migration process for the storage pool, some files that do not satisfy the migration delay time may be migrated unnecessarily. As one process migrates files that satisfy the migration delay time, a second process could begin migrating files that do not satisfy the migration delay time to meet the low migration threshold. The first process that is still migrating files that satisfy the migration delay time might have, by itself, caused the low migration threshold to be met.

### No

Specifies that the server stops migration when no eligible files remain to be migrated, even before reaching the low migration threshold. The server does not migrate files unless the files satisfy the migration delay time.

## MIGProcess

Specifies the number of parallel processes to use for migrating the files from the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999. The default value is 1.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the migration, and the number of logical and physical drives that can be dedicated to the operation. To access a sequential-access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the migration.

For example, suppose you want to simultaneously migrate the files from volumes in two primary sequential storage pools and that you want to specify three processes for each of the storage pools. The storage pools have the same device class. Assuming that the storage pool to which files are being migrated

## DEFINE STGPOOL

has the same device class as the storage pool from which files are being migrated, each process requires two mount points and, if the device type is not FILE, two drives. (One drive is for the input volume, and the other drive is for the output volume.) To run six migration processes simultaneously, you need a total of at least 12 mount points and 12 drives. The device class for the storage pools must have a mount limit of at least 12.

If the number of migration processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the migration processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of migration processes regardless of the number of volumes that are eligible for migration. For example, if you specify ten migration processes and only six volumes are eligible for migration, the server will start ten processes and four of them will complete without processing a volume.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### DATAFormat

Specifies the data format to use to back up files to this storage pool and restore files from this storage pool. The default format is the NATIVE server format. Possible values are:

#### NATive

Specifies the data format is the native IBM Tivoli Storage Manager server format and includes block headers.

#### NONblock

Specifies the data format is the native IBM Tivoli Storage Manager server format and does not include block headers.

**Important:** The default minimum block size on a volume associated with a FILE device class is 256 KB, regardless how much data is being written to the volume. For certain tasks (for example, using content-management products, using the DIRMC client option to store directory information, or migrating very small files using the Tivoli Storage Manager for Space Management client), you can minimize wasted space on storage volumes by specifying the NONBLOCK data format. In most situations, however, the NATIVE format is preferred.

### NETAPPDump

Specifies the data is in a NetApp dump format. This data format should be specified for file system images that are in a dump format and that have been backed up from a NetApp or an IBM System Storage N Series file server using NDMP. The server will not perform migration, reclamation, or AUDIT VOLUME for a storage pool with DATAFORMAT=NETAPPDUMP. You can use the MOVE DATA command to move data from one primary storage pool to another, or out of a volume if the volume needs to be reused.



### **CELERRADump**

Specifies that the data is in an EMC Celerra dump format. This data format should be specified for file system images that are in a dump format and that have been backed up from an EMC Celerra file server using NDMP. The server will not perform migration, reclamation, or AUDIT VOLUME for a storage pool with DATAFORMAT=CELERRADUMP. You can use the MOVE DATA command to move data from one primary storage pool to another, or out of a volume if the volume needs to be reused.

### **NDMPDump**

Specifies that the data is in NAS vendor-specific backup format. Use this data format for file system images that have been backed up from a NAS file server other than a NetApp or EMC Celerra file server. The server will not perform migration, reclamation, or AUDIT VOLUME for a storage pool with DATAFORMAT=NDMPDUMP. You can use the MOVE DATA command to move data from one primary storage pool to another, or out of a volume if the volume needs to be reused.

### **COPYSTGpools**

Specifies the names of copy storage pools where the server simultaneously writes data. The COPYSTGPOOLS parameter is optional. You can specify a maximum of three copy pool names separated by commas. (In versions earlier than Version 5 Release 3, the maximum number was ten.) Spaces between the names of the copy pools are not permitted. When specifying a value for the COPYSTGPOOLS parameter, you can also specify a value for the COPYCONTINUE parameter.

The combined total number of storage pools specified in the COPYSGTPOOLS and ACTIVEDATAPOOLS parameters cannot exceed three.

The simultaneous write function changed in Tivoli Storage Manager Version 5.3. In Version 5.3 and later versions, when a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of copy storage pools and the COPYCONTINUE value from the primary storage pool. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to copy storage pools for the following operations:

- Backup and archive operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API
- Migration operations by Tivoli Storage Manager for Space Management clients
- Import operations that involve copying exported file data from external media to a storage pool defined with a copy storage pool list

### **Restrictions:**

1. This parameter is available only to primary storage pools that use NATIVE or NONBLOCK data format. This parameter is not available for storage pools that use the following data formats:
  - NETAPPDUMP
  - CELERRADUMP
  - NDMPDUMP
2. Simultaneous write to copy storage pools is not supported when the operation is using LAN-free data movement. Simultaneous write takes

## DEFINE STGPOOL

precedence over LAN-free operations, causing the operations to go over the LAN. However, the simultaneous write configuration is honored.

3. Simultaneous write is not supported for NAS backup operations. If the primary storage pool specified in the DESTINATION or TOCDESTINATION in the copy group of the management class has copy storage pools defined, the copy storage pools are ignored and the data is stored into the primary storage pool only.
4. You cannot use the simultaneous write function with Centera storage devices.

**Attention:** The function provided by the COPYSTGPOLLS parameter is not intended to replace the BACKUP STGPOOL command. If you use the COPYSTGPOLLS parameter, continue to use the BACKUP STGPOOL command to ensure that the copy storage pools are complete copies of the primary storage pool. There are cases when a copy may not be created. For more information, see the COPYCONTINUE parameter description.

### COPYContinue

Specifies how the server should react to a copy storage pool write failure for any of the copy storage pools listed in the COPYSTGPOLLS parameter. This parameter is optional. The default value is YES. When specifying the COPYCONTINUE parameter, you must also specify the COPYSTGPOLLS parameter.

This parameter has no effect on active-data pools. If a write error occurs while writing to an active-data pool, the server will stop writing to the failing active-data pool for the remainder of the session.

Possible values are:

#### Yes

If the COPYCONTINUE option is set to YES, the server will stop writing to the failing copy pools for the remainder of the session, but continue storing files into the primary pool and any remaining copy pools. The copy storage pool list is active only for the life of the client session and applies to all the primary storage pools in a particular storage pool hierarchy.

For additional information about the COPYCONTINUE parameter, refer to the information about simultaneous write in the *Administrator's Guide*.

#### No

If the COPYCONTINUE option is set to NO, the server will fail the current transaction and discontinue the store operation.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### ACTIVEDATApools

Specifies the names of active-data pools where the server simultaneously writes data during a client backup operation. The ACTIVEDATAPOOLS parameter is optional. Spaces between the names of the active-data pools are not permitted.

The combined total number of storage pools specified in the COPYSGTPOOLS and ACTIVEDATAPOOLS parameters can not exceed three.



When a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of active-data pools from the destination storage pool specified in the copy group. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API.

## Restrictions:

1. This parameter is available only to primary storage pools that use NATIVE or NONBLOCK data format. This parameter is not available for storage pools that use the following data formats:
  - NETAPPDUMP
  - CELERRADUMP
  - NDMPDUMP
2. Simultaneous write to active-data pools is not supported when the operation is using LAN-free data movement. Simultaneous write takes precedence over LAN-free operations, causing the operations to go over the LAN. However, the simultaneous write configuration is honored.
3. Simultaneous write is not supported when a NAS backup operation is writing a TOC file. If the primary storage pool specified in the TOCDESTINATION in the copy group of the management class has active-data pools defined, the active-data pools are ignored, and the data is stored into the primary storage pool only.
4. You cannot use the simultaneous write function with Centera storage devices.
5. Data being imported will not be stored in active-data pools. After an import operation, use the COPY ACTIVATEDATA command to store the imported data in an active-data pool.

**Attention:** The function provided by the ACTIVEDATAPOOLS parameter is not intended to replace the COPY ACTIVATEDATA command. If you use the ACTIVEDATAPOOLS parameter, use the COPY ACTIVATEDATA command to ensure that the active-data pools contain all active data of the primary storage pool.

## DEDuplicate

Specifies whether the data that is stored in this storage pool will be deduplicated. This parameter is optional and is valid only for storage pools that are defined with a FILE-type device class. The default value is NO.

## IDENTIFYPRocess

Specifies the number of parallel processes to use for duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value from 0 to 20. The default value is 1. If the value of the parameter DEDuplicate is NO, the default setting for IDENTIFYPROCESS has no effect.

When calculating the value for this parameter, consider the workload on the server and the amount of data requiring deduplication. Deduplication identification requires disk I/O and CPU resources, so the more processes you allocate to deduplication, the heavier the workload you place on your system. In addition, consider the number of volumes that require processing.

## DEFINE STGPOOL

Duplicate-identification processes work on volumes containing data that requires deduplication. If you update a storage pool, specifying that the data in the storage pool is to be deduplicated, all of the volumes in the pool will require processing. For this reason, you might need to define a high number of duplicate-identification processes initially. Over time, however, as existing volumes are processed, only the volumes containing new data will need to be processed. At this time, you can reduce the number of duplicate-identification processes.

**Important:** Duplicate-identification processes can be either active or idle. Processes that are working on files are active. Processes that are waiting for files to work on are idle. Processes remain idle until volumes with data to be deduplicated become available. The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. (For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.) Processes end only when cancelled or when the number of duplicate-identification processes for the storage pool is changed to a value less than that currently specified.

### **Example: Define a primary storage pool with an 8MMTAPE device class**

Define a primary storage pool named 8MMPPOOL to the 8MMTAPE device class (with a device type of 8MM) with a maximum file size of 5 MB. Store any files larger than 5 MB in subordinate pools, beginning with POOL1. Enable collocation of files for client nodes. Allow as many as 5 scratch volumes for this storage pool.

```
define stgpool 8mmpool 8mmtape maxsize=5m
nextstgpool=pool1 collocate=node
maxscratch=5
```

## DEFINE STGPOOL (Define a copy storage pool assigned to sequential access devices)

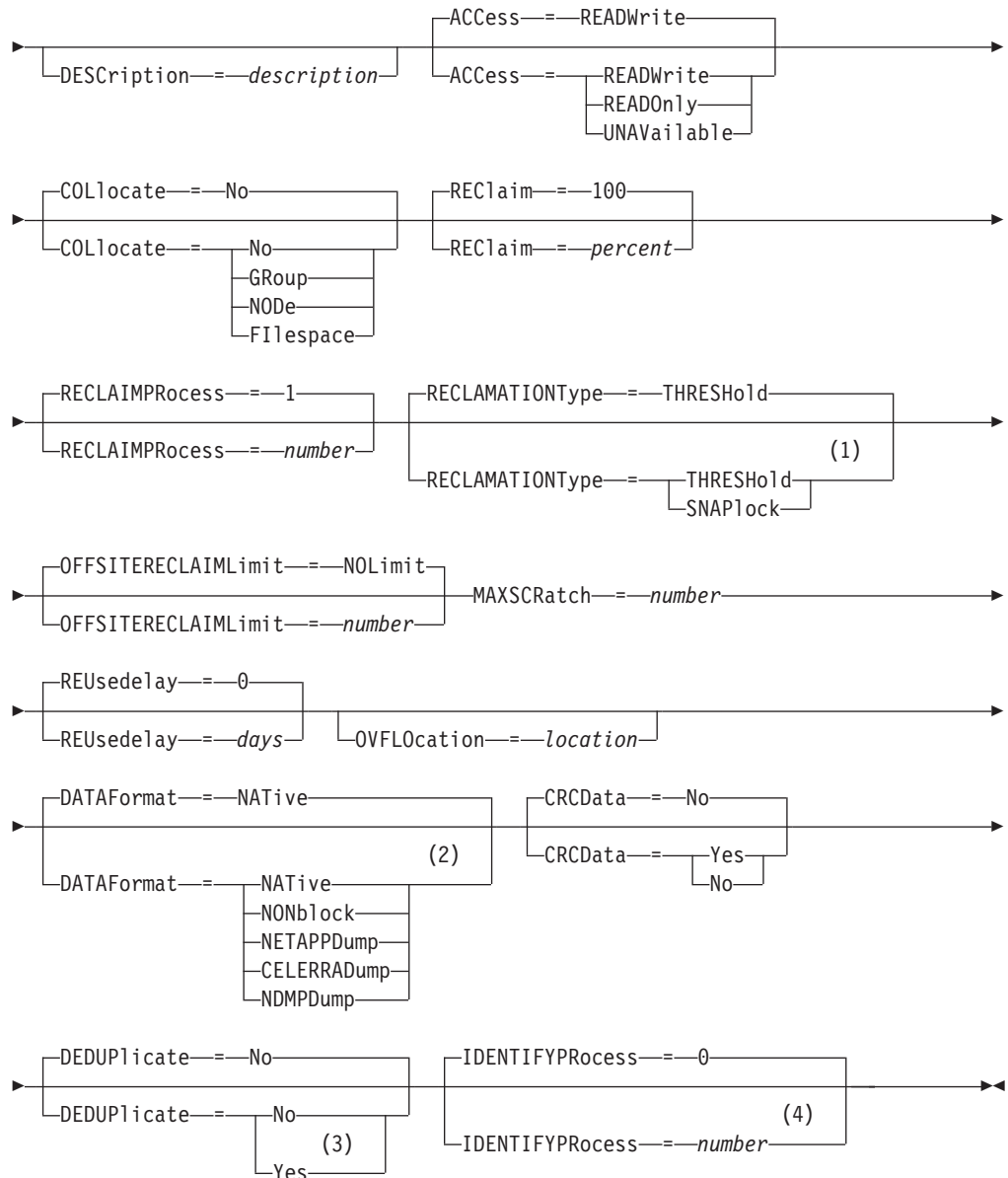
Use this command to define a copy storage pool assigned to sequential access devices.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► DEFINE STGpool *pool\_name* *device\_class\_name* POOLtype == COPY ►►



### Notes:

- 1 The `RECLAMATIONTYPE=SNAPLOCK` setting is valid only for storage

## DEFINE STGPOOL

pools defined to servers that are enabled for System Storage Archive Manager. The storage pool must be assigned to a FILE device class, and the directories specified in the device class must be NetApp SnapLock volumes.

- 2 The values NETAPPDUMP, CELERRADUMP, and NDMPDUMP are not valid for storage pools that are defined with a FILE device class.
- 3 This parameter is valid only for storage pools that are defined with a FILE device class.
- 4 This parameter is available only when the value of the DEDUPLICATE parameter is YES.

### Parameters

#### *pool\_name* (Required)

Specifies the name of the storage pool to be defined. The name must be unique, and the maximum length is 30 characters.

#### *device\_class\_name* (Required)

Specifies the name of the sequential access device class to which this copy storage pool is assigned. You can specify any device class except DISK.

#### **POoltype=COpy** (Required)

Specifies that you want to define a copy storage pool.

### DESCription

Specifies a description of the copy storage pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

### ACCess

Specifies how client nodes and server processes (such as 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 client nodes can only read files stored on the volumes in the copy storage pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the copy storage pool to restore files to primary storage pools. However, no new writes are permitted to volumes in the copy storage pool from volumes outside the storage pool. A storage pool cannot be backed up to the copy storage pool.

#### **UNAVailable**

Specifies that client nodes cannot access files stored on volumes in the copy storage pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the copy storage pool to restore files to primary storage pools. However, no new writes are permitted to volumes in the copy storage pool from volumes outside the storage pool. A storage pool cannot be backed up to the copy storage pool.

### COLlocate

Specifies whether the server attempts to keep data belonging to a single client

node, group of client nodes, or client file space stored on 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. For details, see the *Administrator's Guide*.

Possible values are:

## No

Specifies that collocation is disabled.

## GRoup

Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible. If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

If you specify COLLOCATE=GROUP but do not define any collocation groups or if you specify COLLOCATE=GROUP but do not add nodes to a collocation group, data is collocated by node. Be sure to consider tape usage when organizing client nodes into collocation groups. For example, if a tape-based storage pool consists of data from grouped and ungrouped nodes and you specify COLLOCATE=GROUP, the server performs the following actions:

- Collocates by group the data for grouped nodes only. Whenever possible, the server collocates data belonging to a group of nodes on a single tape or on as few tapes as possible. Data for a single node can also be spread across several tapes associated with a group.
- Collocates by node the data for ungrouped nodes. Whenever possible, the server stores the data for a single node on a single tape. All available tapes that already have data for the node are used before available space on any other tape is used.

## NODe

Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible. If the node has multiple file spaces, the server does not attempt to collocate those file spaces. For backward compatibility, COLLOCATE=YES is still accepted by the server to specify collocation at the client node level.

If a storage pool contains data for a node that is a member of a collocation group and you specify COLLOCATE=NODE, the data will be collocated by node not by group.

## Filespace

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

## REClaim

Specifies when the server reclaims a volume, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space on volumes usable again by moving any remaining unexpired 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 to 100. The default value is 100, which means that reclamation is not performed.

## DEFINE STGPOOL

If you change the value from the default, specify a value of 50 percent or greater so that files stored on two volumes can be combined onto a single output volume.

When a copy pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to obtain the unexpired files on the reclaimable volume from a primary or copy storage pool that is onsite. The process then writes these files to an available volume in the original copy storage pool. Effectively, these files are moved back to the onsite location. However, the files could be obtained from the offsite volume after a disaster if a database backup is used that references the files on the offsite volume. Because of the way reclamation works with offsite volumes, use it carefully with copy storage pools.

### **RECLAIMPRocess**

Specifies the number of parallel processes to use for reclaiming the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999. The default value is 1.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the reclamation and the number of logical and physical drives that can be dedicated to the operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the reclamation.

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that you want to specify four processes for each of the storage pools. The storage pools have the same device class. Each process requires two mount points and, if the device type is not FILE, two drives. (One of the drives is for the input volume, and the other drive is for the output volume.) To run eight reclamation processes simultaneously, you need a total of at least 16 mount points and 16 drives. The device class for the storage pools must have a mount limit of at least 16.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the reclamation processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

### **RECLAMATIONType**

Specifies the method by which volumes are reclaimed and managed. This parameter is optional. The default value is THRESHOLD. Possible values are the following:

#### **THRESHold**

Specifies that volumes belonging to this storage pool are reclaimed based on the threshold value in the RECLAIM attribute for this storage pool.

### SNAPlock

Specifies that FILE volumes belonging to this storage pool will be managed for retention using NetApp Data ONTAP software and NetApp SnapLock volumes. This parameter is only valid for storage pools being defined to a server that has data retention protection enabled and that is assigned to a FILE device class. Volumes in this storage pool are not reclaimed based on threshold; the RECLAIM value for the storage pool is ignored.

All volumes in this storage pool are created as FILE volumes. A retention date, derived from the retention attributes in the archive copy group for the storage pool, is set in the metadata for the FILE volume using the SnapLock feature of the NetApp Data ONTAP operating system. Until the retention date has expired, the FILE volume and any data on it cannot be deleted from the physical SnapLock volume on which it is stored.

The RECLAMATIONTYPE parameter for all storage pools being defined must be the same when defined to the same device class name. The DEFINE command will fail if the RECLAMATIONTYPE parameter specified is different than what is currently defined for storage pools already defined to the device class name.

### OFFSITERECLAIMLimit

Specifies the number of offsite volumes to have their space reclaimed during reclamation for this storage pool. This parameter is optional. The default value is NOLIMIT. Possible values are:

#### NOLimit

Specifies that you want to have the space reclaimed in all of your offsite volumes.

#### *number*

Specifies the number of offsite volumes to have their space reclaimed. You can specify an integer from 0 to 99999. A value of zero means that none of the offsite volumes will be reclaimed.

**Important:** When determining the value for the OFFSITERECLAIMLIMIT, consider using the statistical information in the message issued at the end of the offsite volume reclamation operation. Alternatively, you can use the following Tivoli Storage Manager SQL select command to obtain the statistical information from the SUMMARY table for the offsite volume reclamation operation:

```
select * from summary where activity='OFFSITE RECLAMATION'
```

The statistical information includes the following items:

- The number of offsite volumes that were processed
- The number of parallel processes that were used
- The total amount of time required for the processing

The order in which offsite volumes are reclaimed is based on the amount of unused space in a volume. (Unused space includes both space that has never been used on the volume and space that has become empty because of file deletion.) Volumes with the largest amount of unused space are reclaimed first.

For example, suppose a copy storage pool contains three volumes: VOL1, VOL2, and VOL3. VOL1 has the largest amount of unused space, and VOL3 has the least amount of unused space. Suppose further that the percentage of unused space in each of the three volumes is greater than the value of the RECLAIM parameter. If you do not specify a value for the



## DEFINE STGPOOL

OFFSITERECLAIMLIMIT parameter, all three volumes will be reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 will be reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 will be reclaimed.

### MAXSCRatch (Required)

Specifies the maximum number of scratch volumes that the server can request for this storage pool. You can specify an integer from 0 to 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

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.

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 is not deleted from the copy storage pool until the access mode is changed. This allows an administrator to query the server 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 the server and returned to the file system.

**Tip:** For server-to-server operations that utilize virtual volumes and that store a small amount of data, consider specifying a value for the MAXSCRATCH parameter that is higher than the value you typically specify for write operations to other types of volumes. After a write operation to a virtual volume, Tivoli Storage Manager marks the volume as FULL, even if the value of the MAXCAPACITY parameter on the device-class definition has not been reached. The Tivoli Storage Manager server does not keep virtual volumes in FILLING status and does not append to them. If the value of the MAXSCRATCH parameter is too low, server-to-server operations can fail.

### REUsedelay

Specifies the number of days that must elapse after all files are 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 to 9999. The default value is 0, which means that a volume can be rewritten or returned to the scratch pool as soon as all the files are deleted from the volume.

**Important:** Use this parameter to help ensure that when you restore the database to an earlier level, database references to files in the copy storage pool are still valid. You must set this parameter to a value greater than the number of days you plan to retain the oldest database backup. The number of days specified for this parameter should be the same as the number specified for the SET DRMDBBACKUPEXPIREDAYS command. For more information, see the *Administrator's Guide*.

### OVFLOcation

Specifies the overflow location for the storage pool. The server assigns this location name to a volume that is ejected from the library by the command. This parameter is optional. The location name can be a maximum length of 255 characters. Enclose the location name in quotation marks if the location name contains any blank characters.



### **DATAFormat**

Specifies the data format to use to back up files to this storage pool and restore files from this storage pool. The default format is the NATIVE server format. Possible values are:

#### **NATive**

Specifies the data format is the native IBM Tivoli Storage Manager server format and includes block headers.

#### **NONblock**

Specifies the data format is the native IBM Tivoli Storage Manager server format and does not include block headers.

**Tip:** The default minimum block size on a volume associated with a FILE device class is 256 KB, regardless how much data is being written to the volume. For certain tasks (for example, using content-management products, using the DIRMC client option to store directory information, or migrating very small files using the Tivoli Storage Manager for Space Management client), you can minimize wasted space on storage volumes by specifying the NONBLOCK data format. In most situations, however, the NATIVE format is preferred.

### **NETAPPDump**

Specifies that the data is in a NetApp dump format. Do not specify this data format for file system images that are in a dump format and that have been backed up from a NetApp file server using NDMP. The server will not perform storage pool reclamation or AUDIT VOLUME for a storage pool with DATAFORMAT=NETAPPDUMP. You can use the MOVE DATA command to move NDMP-generated data out of a volume if the volume needs to be re-used.

### **CELERRADump**

Specifies that the data is in an EMC Celerra dump format. Do not specify this data format for file system images that are in a dump format and that have been backed up from an EMC Celerra file server using NDMP. The server will not perform storage pool reclamation or AUDIT VOLUME for a storage pool with DATAFORMAT=CELERRADUMP. You can use the MOVE DATA command to move NDMP-generated data out of a volume if the volume needs to be re-used.

### **NDMPDump**

Specifies that the data is in a NAS vendor-specific backup format. Do not specify this data format for file system images that are in a backup format and that have been backed up from a NAS file server other than a NetApp or EMC Celerra file server. The server will not perform storage pool reclamation or AUDIT VOLUME for a storage pool with DATAFORMAT=NDMPDUMP. You can use the MOVE DATA command to move NDMP-generated data out of a volume if the volume needs to be re-used.

### **CRCData**

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is only valid for NATIVE data format storage pools. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

## DEFINE STGPOOL

### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

### No

Specifies that data is stored without CRC information.

### DEDuplicate

Specifies whether the data that is stored in this storage pool will be deduplicated. This parameter is optional and is valid only for storage pools that are defined with a FILE-type device class. The default value is NO.

### IDENTIFYProcess

Specifies the number of parallel processes to use for duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value from 0 to 20.

The default value for this parameter is 0. Duplicate-identification processes for a copy storage pool are not necessary if you specify duplicate-identification processes for the primary storage pool. When Tivoli Storage Manager analyzes a file in a storage pool, Tivoli Storage Manager also analyzes the file in all other storage pools.

When calculating the value for this parameter, consider the workload on the server and the amount of data requiring deduplication. Deduplication identification requires disk I/O and CPU resources, so the more processes you allocate to deduplication, the heavier the workload you place on your system. In addition, consider the number of volumes that require processing. Duplicate-identification processes work on volumes containing data that requires deduplication. If you update a storage pool, specifying that the data in the storage pool is to be deduplicated, all of the volumes in the pool will require processing. For this reason, you might need to define a high number of duplicate-identification processes initially. Over time, however, as existing volumes are processed, only the volumes containing new data will need to be processed. At this time, you can reduce the number of duplicate-identification processes.

**Important:** Duplicate-identification processes can be either active or idle. Processes that are working on files are active. Processes that are waiting for files to work on are idle. Processes remain idle until volumes with data to be deduplicated become available. The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. (For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.) Processes end only when cancelled or when the number of duplicate-identification processes for the storage pool is changed to a value less than that currently specified.

### Example: Define a copy storage pool with a DC480 device class.

Define a copy storage pool, TAPEPOOL2, to the DC480 device class. Allow up to 50 scratch volumes for this pool. Delay the reuse of volumes for 45 days.

```
define stgpool tapepool2 dc480 pooltype=copy
maxscratch=50 reusedelay=45
```

## DEFINE STGPOOL (Define an active-data pool assigned to sequential-access devices)

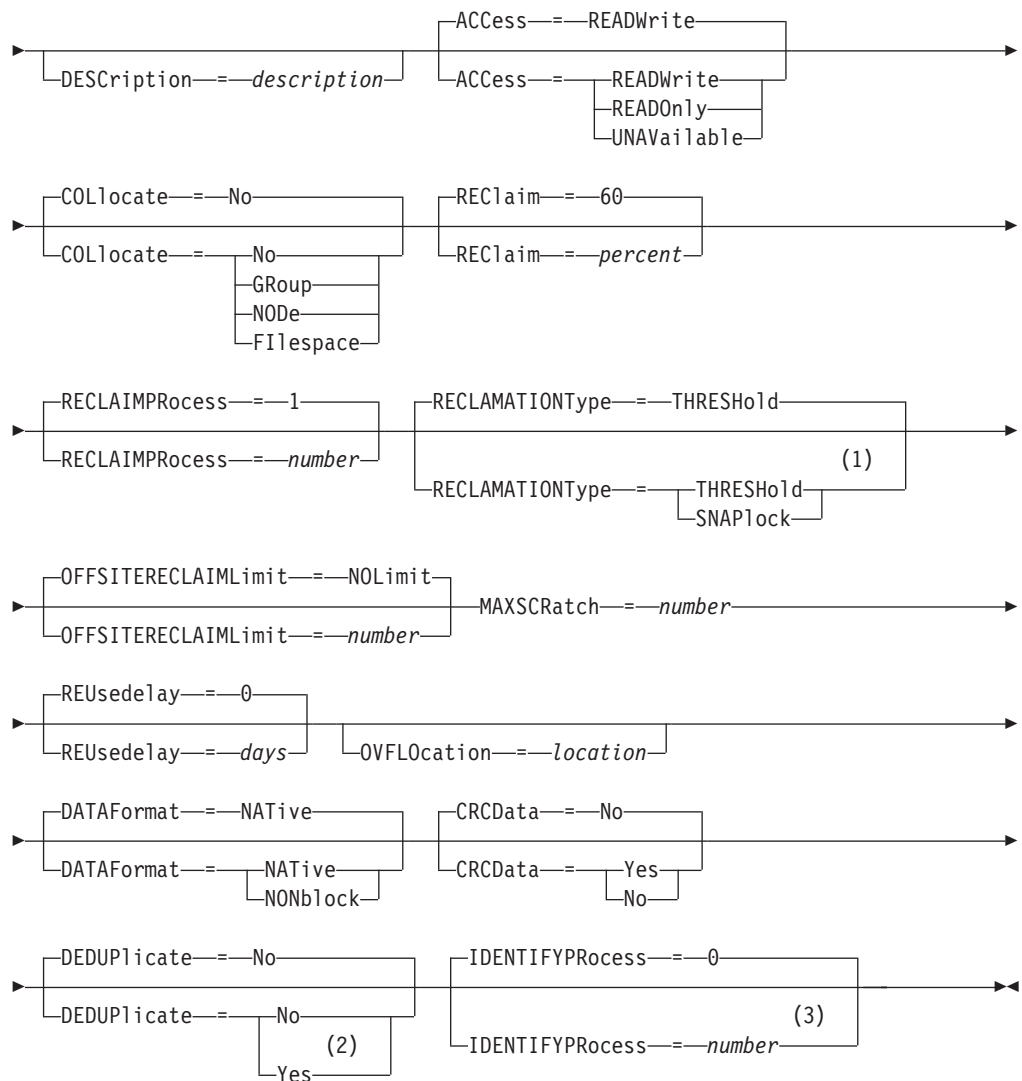
Use this command to define an active-data pool assigned to sequential-access devices.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► Define STGpool *pool\_name* *device\_class\_name* POOLtype == ACTIVEdata ►►



### Notes:

- 1 The RECLAMATIONTYPE=SNAPLOCK setting is valid only for storage pools defined to servers that are enabled for System Storage Archive Manager. The storage pool must be assigned to a FILE device class, and the directories specified in the device class must be NetApp SnapLock volumes.

## DEFINE STGPOOL

- 2 This parameter is valid only for storage pools that are defined with a FILE device class.
- 3 This parameter is available only when the value of the DEDUPLICATE parameter is YES.

### Parameters

#### *pool\_name* (Required)

Specifies the name of the storage pool to be defined. The name must be unique, and the maximum length is 30 characters.

#### *device\_class\_name* (Required)

Specifies the name of the sequential access device class to which this active-data pool is assigned. You can specify any device class except DISK.

#### **POOLtype=ACTIVEdata** (Required)

Specifies that you want to define an active-data pool.

### DESCRIPTION

Specifies a description of the active-data pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

### ACCESS

Specifies how client nodes and server processes (such as reclamation) can access files in the active-data 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 active-data pool.

#### **READOnly**

Specifies that client nodes can only read files stored on the volumes in the active-data pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the active-data pool to restore files to primary storage pools. However, no new writes are permitted to volumes in the active-data pool from volumes outside the storage pool. A storage pool cannot be copied to the active-data pool.

#### **UNAVAILABLE**

Specifies that client nodes cannot access files stored on volumes in the active-data pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the active-data pool to restore files to primary storage pools. However, no new writes are permitted to volumes in the active-data pool from volumes outside the storage pool. A storage pool cannot be copied to the active-data pool.

### COLLOCATE

Specifies whether the server attempts to keep data belonging to a single client node, group of client nodes, or client file space stored on 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. For details, see the *Administrator's Guide*.

Possible values are:

**No**

Specifies that collocation is disabled.

**GRoup**

Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible. If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

If you specify COLLOCATE=GROUP but do not define any collocation groups or if you specify COLLOCATE=GROUP but do not add nodes to a collocation group, data is collocated by node. Be sure to consider tape usage when organizing client nodes into collocation groups. For example, if a tape-based storage pool consists of data from grouped and ungrouped nodes and you specify COLLOCATE=GROUP, the server performs the following actions:

- Collocates by group the data for grouped nodes only. Whenever possible, the server collocates data belonging to a group of nodes on a single tape or on as few tapes as possible. Data for a single node can also be spread across several tapes associated with a group.
- Collocates by node the data for ungrouped nodes. Whenever possible, the server stores the data for a single node on a single tape. All available tapes that already have data for the node are used before available space on any other tape is used.

**NODe**

Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible. If the node has multiple file spaces, the server does not attempt to collocate those file spaces. For backward compatibility, COLLOCATE=YES is still accepted by the server to specify collocation at the client node level.

If a storage pool contains data for a node that is a member of a collocation group and you specify COLLOCATE=NODE, the data will be collocated by node not by group.

**Filespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

**REClaim**

Specifies when the server reclaims a volume, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space and space occupied by inactive backup files on volumes usable again by moving any remaining unexpired files and active backup files from one volume to another volume. This makes the original volume available for reuse. This parameter is optional. You can specify an integer from 1 to 100. The default value is 60.

If you change the value from the default, specify a value of 50 percent or greater so that files stored on two volumes can be combined onto a single output volume.

When an active-data pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to obtain the unexpired files on

## DEFINE STGPOOL

the reclaimable volume from a primary or active-data pool that is onsite. The process then writes these files to an available volume in the original active-data pool. Effectively, these files are moved back to the onsite location. However, the files could be obtained from the offsite volume after a disaster if a database backup is used that references the files on the offsite volume. Because of the way reclamation works with offsite volumes, use it carefully with active-data pools.

### **RECLAIMPRocess**

Specifies the number of parallel processes to use for reclaiming the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999. The default value is 1.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the reclamation and the number of logical and physical drives that can be dedicated to the operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the reclamation.

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that you want to specify four processes for each of the storage pools. The storage pools have the same device class. Each process requires two mount points and, if the device type is not FILE, two drives. (One of the drives is for the input volume, and the other drive is for the output volume.) To run eight reclamation processes simultaneously, you need a total of at least 16 mount points and 16 drives. The device class for the storage pools must have a mount limit of at least 16.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the reclamation processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

### **RECLAMATIONType**

Specifies the method by which volumes are reclaimed and managed. This parameter is optional. The default value is THRESHOLD. Possible values are the following:

#### **THRESHold**

Specifies that volumes belonging to this storage pool are reclaimed based on the threshold value in the RECLAIM attribute for this storage pool.

#### **SNAPlock**

Specifies that FILE volumes belonging to this storage pool will be managed for retention using NetApp Data ONTAP software and NetApp SnapLock volumes. This parameter is only valid for storage pools being defined to a server that has data retention protection enabled and that is assigned to a

FILE device class. Volumes in this storage pool are not reclaimed based on threshold; the RECLAIM value for the storage pool is ignored.

All volumes in this storage pool are created as FILE volumes. A retention date, derived from the retention attributes in the archive copy group for the storage pool, is set in the metadata for the FILE volume using the SnapLock feature of the NetApp Data ONTAP operating system. Until the retention date has expired, the FILE volume and any data on it cannot be deleted from the physical SnapLock volume on which it is stored.

The RECLAMATIONTYPE parameter for all storage pools being defined must be the same when defined to the same device class name. The DEFINE command will fail if the RECLAMATIONTYPE parameter specified is different than what is currently defined for storage pools already defined to the device class name.

## OFFSITERECLAIMLimit

Specifies the number of offsite volumes to have their space reclaimed during reclamation for this storage pool. This parameter is optional. The default value is NOLIMIT. Possible values are:

### NOLimit

Specifies that you want to have the space reclaimed in all of your offsite volumes.

### *number*

Specifies the number of offsite volumes to have their space reclaimed. You can specify an integer from 0 to 99999. A value of zero means that none of the offsite volumes will be reclaimed.

**Important:** When determining the value for the OFFSITERECLAIMLIMIT, consider using the statistical information in the message issued at the end of the offsite volume reclamation operation. Alternatively, you can use the following Tivoli Storage Manager SQL select command to obtain the statistical information from the SUMMARY table for the offsite volume reclamation operation:

```
select * from summary where activity='OFFSITE RECLAMATION'
```

The statistical information includes the following items:

- The number of offsite volumes that were processed
- The number of parallel processes that were used
- The total amount of time required for the processing

The order in which offsite volumes are reclaimed is based on the amount of unused space in a volume. (Unused space includes both space that has never been used on the volume and space that has become empty because of file deletion.) Volumes with the largest amount of unused space are reclaimed first.

For example, suppose an active-data pool contains three volumes: VOL1, VOL2, and VOL3. VOL1 has the largest amount of unused space, and VOL3 has the least amount of unused space. Suppose further that the percentage of unused space in each of the three volumes is greater than the value of the RECLAIM parameter. If you do not specify a value for the OFFSITERECLAIMLIMIT parameter, all three volumes will be reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 will be reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 will be reclaimed.



### MAXSCRatch (Required)

Specifies the maximum number of scratch volumes that the server can request for this storage pool. You can specify an integer from 0 to 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

The value specified for this parameter is used to estimate the total number of volumes available in the active-data pool and the corresponding estimated capacity for the active-data pool.

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 is not deleted from the active-data pool until the access mode is changed. This allows an administrator to query the server 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 the server and returned to the file system.

**Tip:** For server-to-server operations that utilize virtual volumes and that store a small amount of data, consider specifying a value for the MAXSCRATCH parameter that is higher than the value you typically specify for write operations to other types of volumes. After a write operation to a virtual volume, Tivoli Storage Manager marks the volume as FULL, even if the value of the MAXCAPACITY parameter on the device-class definition has not been reached. The Tivoli Storage Manager server does not keep virtual volumes in FILLING status and does not append to them. If the value of the MAXSCRATCH parameter is too low, server-to-server operations can fail.

### REUsedelay

Specifies the number of days that must elapse after all files are 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 to 9999. The default value is 0, which means that a volume can be rewritten or returned to the scratch pool as soon as all the files are deleted from the volume.

**Important:** Use this parameter to help ensure that when you restore the database to an earlier level, database references to files in the active-data pool are still valid. You must set this parameter to a value greater than the number of days you plan to retain the oldest database backup. The number of days specified for this parameter should be the same as the number specified for the SET DRMDBBACKUPEXPIREDAYS command. For more information, see the *Administrator's Guide*.

### OVFLOcation

Specifies the overflow location for the storage pool. The server assigns this location name to a volume that is ejected from the library by the command. This parameter is optional. The location name can be a maximum length of 255 characters. Enclose the location name in quotation marks if the location name contains any blank characters.

### DATAFormat

Specifies the data format to use to copy files to this storage pool and restore files from this storage pool. The default format is the NATIVE server format. Possible values are:



## NATive

Specifies the data format is the native IBM Tivoli Storage Manager server format and includes block headers.

## NONblock

Specifies the data format is the native IBM Tivoli Storage Manager server format and does not include block headers.

**Important:** The default minimum block size on a volume associated with a FILE device class is 256 KB, regardless how much data is being written to the volume. For certain tasks (for example, using content-management products, using the DIRMC client option to store directory information, or migrating very small files using the Tivoli Storage Manager for Space Management client), you can minimize wasted space on storage volumes by specifying the NONBLOCK data format. In most situations, however, the NATIVE format is preferred.

## CRCData

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is only valid for NATIVE data format storage pools. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

### No

Specifies that data is stored without CRC information.

## DEDuplicate

Specifies whether the data that is stored in this storage pool will be deduplicated. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. The default value is NO.

## IDENTIFYProcess

Specifies the number of parallel processes to use for duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value from 0 to 20.

The default value for this parameter is 0. Duplicate-identification processes for a copy storage pool are not necessary if you specify duplicate-identification processes for the primary storage pool. When Tivoli Storage Manager analyzes a file in a storage pool, Tivoli Storage Manager also analyzes the file in all other storage pools.

When calculating the value for this parameter, consider the workload on the server and the amount of data requiring deduplication. Deduplication identification requires disk I/O and CPU resources, so the more processes you allocate to deduplication, the heavier the workload you place on your system. In addition, consider the number of volumes that require processing. Duplicate-identification processes work on volumes containing data that requires deduplication. If you update a storage pool, specifying that the data in the storage pool is to be deduplicated, all of the volumes in the pool will require processing. For this reason, you might need to define a high number of duplicate-identification processes initially. Over time, however, as existing

## DEFINE STGPOOL

volumes are processed, only the volumes containing new data will need to be processed. At this time, you can reduce the number of duplicate-identification processes.

**Important:** Duplicate-identification processes can be either active or idle. Processes that are working on files are active. Processes that are waiting for files to work on are idle. Processes remain idle until volumes with data to be deduplicated become available. The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. (For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.) Processes end only when cancelled or when the number of duplicate-identification processes for the storage pool is changed to a value less than that currently specified.

### Example: Define an active-data pool with a DC500 device class

Define an active-data pool, TAPEPOOL2, to the DC500 device class. Allow up to 50 scratch volumes for this pool. Delay the reuse of volumes for 45 days.

```
define stgpool tapepool3 dc500 pooltype=activedata  
maxscratch=50 reusedelay=45
```

## DEFINE SUBSCRIPTION (Define a profile subscription)

Use this command on a managed server to subscribe that managed server to a profile.

When a server subscribes to its first profile, a subscription is also created to the default profile (if one exists) of the configuration manager. The server then contacts the configuration manager periodically for configuration updates.

### Restrictions:

1. A server cannot subscribe to profiles from more than one configuration manager.
2. If a server subscribes to a profile with an associated object that is already defined on the server, the local definition is replaced by the definition from the configuration manager. For example, if a server has an administrative schedule named WEEKLY\_BACKUP, then subscribes to a profile that also has an administrative schedule named WEEKLY\_BACKUP, the local definition is replaced.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—DEfIne SUBSCRIPtIon—profile_name—[SERVer==server_name]
```

### Parameters

#### *profile\_name* (Required)

Specifies the name of the profile to which the server subscribes.

#### SERVer

Specifies the name of the configuration manager from which the configuration information is obtained. This parameter is required, if the managed server does not have at least one subscription. If the managed server has a subscription, you can omit this parameter and it defaults to the configuration manager for that subscription.

### Example: Define a profile subscription

Subscribe a profile named BETA that resides on a configuration manager named TOM.

```
define subscription beta server=tom
```

### Related commands

Table 107. Commands related to DEFINE SUBSCRIPTION

| Command        | Description  |
|----------------|--|
| COPY PROFILE   | Creates a copy of a profile.                                       |
| DEFINE PROFILE | Defines a profile for distributing information to managed servers. |
| DELETE PROFILE | Deletes a profile from a configuration manager.                    |

## DEFINE SUBSCRIPTION

*Table 107. Commands related to DEFINE SUBSCRIPTION (continued)*

| Command             | Description  |
|---------------------|--|
| DELETE SUBSCRIBER   | Deletes obsolete managed server subscriptions.                                   |
| DELETE SUBSCRIPTION | Deletes a specified profile subscription.  |
| LOCK PROFILE        | Prevents distribution of a configuration profile.                                |
| NOTIFY SUBSCRIBERS  | Notifies servers to refresh their configuration information.                     |
| QUERY PROFILE       | Displays information about configuration profiles.                               |
| QUERY SUBSCRIBER    | Displays information about subscribers and their subscriptions to profiles.      |
| QUERY SUBSCRIPTION  | Displays information about profile subscriptions.                                |
| SET CONFIGREFRESH   | Specifies a time interval for managed servers to contact configuration managers. |
| UNLOCK PROFILE      | Enables a locked profile to be distributed to managed servers.                   |
| UPDATE PROFILE      | Changes the description of a profile.  |

# DEFINE VIRTUALFSMAPPING (Define a virtual file space mapping)

Use this command to define a virtual file space mapping.

Virtual file space names can be used in the NAS data operations BACKUP NODE and RESTORE NODE similar to a file system name. Refer to the documentation about your NAS device for guidance on specifying the parameters for this command.

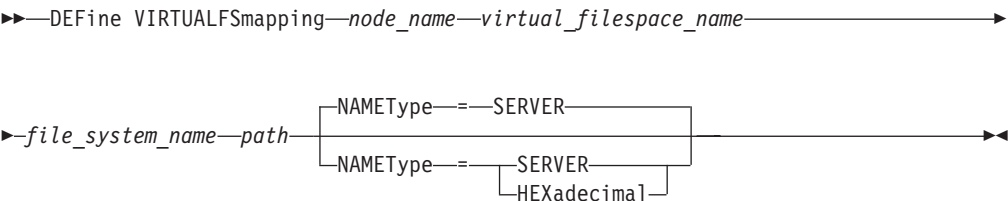
**Note:** The NAS node must have an associated data mover definition because when the Tivoli Storage Manager server updates a virtual file space mapping, the server attempts to contact the NAS device to validate the virtual file system and file system name.

## Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege
- Restricted policy privilege for the domain to which the NAS node is assigned.

## Syntax



## Parameters

### node\_name (Required)

Specifies the NAS node on which the file system and path reside. You cannot use wildcard characters or specify a list of names.

### virtual\_filespace\_name (Required)

Specifies the name which refers to this virtual file space definition. The virtual file space name is case sensitive and the first character must be a forward slash /. The length of the name cannot be more than 64 characters, including the required forward slash. Virtual file space names are restricted to the same character set as all other objects in the Tivoli Storage Manager server except that the forward slash / character is also allowed.

The virtual file space name cannot be identical to any file system on the NAS node. When selecting a virtual file space name, consider the following restrictions:

- If a file system is created on the NAS device with the same name as a virtual file system, a name conflict will occur on the Tivoli Storage Manager server when the new file space is backed up. Use a string for the virtual file space name that is unlikely to be used as a real file system name on your NAS device in the future.

## DEFINE VIRTUALFSMAPPING

For example: A user follows a naming convention for creating file spaces on a NAS device with names of the form /vol1, /vol2, /vol3. The user defines a virtual file space to the Tivoli Storage Manager server with the name /vol9. If the user continues to use the same naming convention, the virtual file space name is likely to conflict with a real file space name at some point in the future.

- During backup and restore operations, Tivoli Storage Manager verifies that a name conflict does not occur prior to starting the operation.
- The virtual file space name appears as a file space in the output of the QUERY FILESPACE command, and also in the backup and restore panels of the Tivoli Storage Manager Web client. Therefore, consider selecting a name that unambiguously identifies this object as a directory path on the NAS device.

### *file\_system\_name* (Required)

Specifies the name of the file system in which the path is located. The file system name must exist on the specified NAS node. The file system name cannot contain wildcard characters.

### *path* (Required)

Specifies the path from the root of the file system to the directory. The path can only reference a directory. The maximum length of the path is 1024 characters. The path name is case sensitive.

### NAMEType

Specifies how the server should interpret the path name specified. This parameter is useful when a path contains characters that are not part of the code page in which the server is running. The default value is SERVER.

Possible values are:

#### SERVER

The server uses the server code page to interpret the path name.

#### HEXadecimal

The server interprets the path that you enter as the hexadecimal representation of the path. This option should be used when a path contains characters that cannot be entered. This could occur if the NAS file system is set to a language different from the one in which the server is running.

## Example: Define a virtual file space mapping

Define the virtual file space mapping name /mikeshomedir for the path /home/mike on the file system /vol/vol1 on the NAS node named NAS1.

```
define virtualfsmapping nas1 /mikeshomedir /vol/vol1 /home/mike
```

## Related commands

Table 108. Commands related to DEFINE VIRTUALFSMAPPING

| Command                 | Description                          |
|-------------------------|--------------------------------------|
| DELETE VIRTUALFSMAPPING | Delete a virtual file space mapping. |
| QUERY VIRTUALFSMAPPING  | Query a virtual file space mapping.  |
| UPDATE VIRTUALFSMAPPING | Update a virtual file space mapping. |

## DEFINE VOLUME (Define a volume in a storage pool)

Use this command to assign a random or sequential access volume to a storage pool.

When defining a random access (DISK) storage pool volume or a sequential access storage pool volume that is associated with a FILE device class, you can have the server create the volume before it is assigned or you can use space triggers to create preassigned volumes when predetermined space-utilization thresholds have been exceeded. (For details about space triggers, see “DEFINE SPACETRIGGER (Define the space trigger)” on page 289.) For volumes associated with device classes other than DISK or device types other than FILE, you can use the DEFINE VOLUME command to assign an already-created volume to a storage pool.

To create space in sequential access storage pools, you can define volumes or allow the server to request scratch volumes as needed, as specified by the MAXSCRATCH parameter for the storage pool. For storage pools associated with the FILE device class, the server can create private volumes as needed using storage pool space triggers. For DISK storage pools, the scratch mechanism is not available. However, you can create space by creating volumes and then defining them to the server or by having the server create volumes using storage pool space triggers.

Tivoli Storage Manager does not validate the existence of a volume name when defining a volume in a storage pool that is associated with a library. The defined volume has “0” EST capacity until data is written to the volume.

### Attention:

- The size of a storage pool volume cannot be changed once it has been defined to the Tivoli Storage Manager server. If you change the size of Tivoli Storage Manager volumes by extending raw logical volumes through SMIT or otherwise altering the file sizes of the volumes with operating system commands or utilities, the server may not initialize correctly and data could be lost.

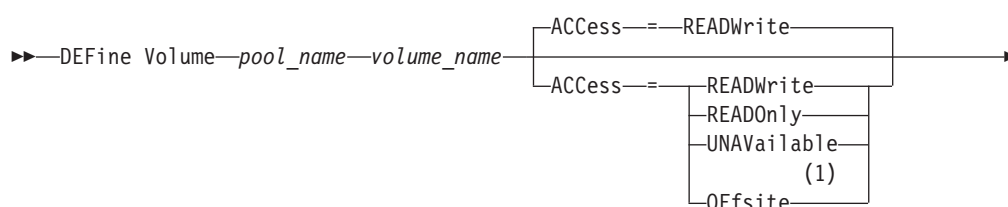
You cannot use this command to define volumes in storage pools with the parameter setting RECLAMATIONTYPE=SNAPLOCK. Volumes in this type of storage pool are allocated by using the MAXSCRATCH parameter on the storage pool definition.

You cannot define volumes in a storage pool defined with the CENTERA device class.

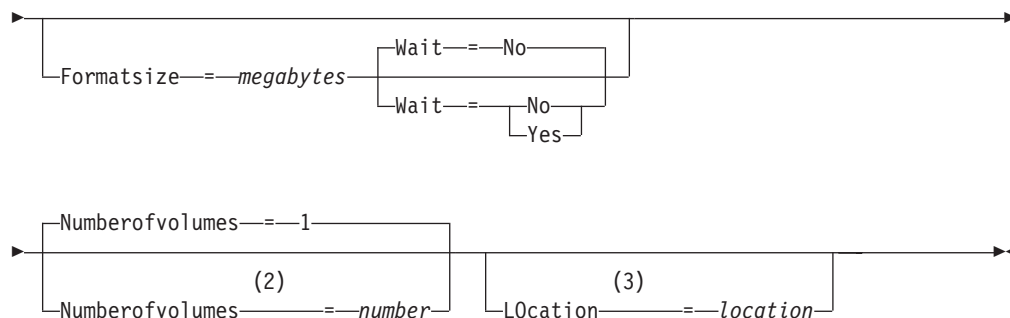
### Privilege class

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



## DEFINE VOLUME



### Notes:

- 1 This value is valid only for volumes assigned to copy storage pools.
- 2 This parameter is valid only for DISK or FILE volumes.
- 3 This parameter is valid only for sequential access volumes.

### Parameters

#### *pool\_name* (Required)

Specifies the name of the storage pool to which the volume is assigned.

#### *volume\_name* (Required)

Specifies the name of the storage pool volume to be defined. If you specify a number greater than 1 for the NUMBEROFVOLUMES parameter, the volume name is used as a prefix to generate multiple volume names. The volume name that you specify depends on the type of device that the storage pool uses. See Table 109.

Each volume used by a server for any purpose must have a unique name. This requirement applies to all volumes, whether the volumes are used for storage pools, or used for operations such as database backup or export. The requirement also applies to volumes that reside in different libraries but that are used by the same server.

**Note:** Volume names cannot contain embedded blanks or equal signs.

Table 109. Volume name requirements by device type

| Device Type | Volume Name Requirements   | Example   |
|-------------|--|---|
| DISK        | The name of the file to contain the volume data, with either the fully qualified path name or a path name relative to the current working directory  | /usr/storage/sbkup01.dsm<br><br>If you are using an AIX logical volume, enter the path name as:<br>/dev/rxxx<br><br>where xxx is the logical volume name. |
| FILE        | The name of the file to contain the volume data, with either the fully qualified path name or the path name relative to a directory identified in the DIRECTORY parameter for the device class.<br><br>Place FILE volumes in one of the directories specified with the DIRECTORY parameter of the DEFINE DEVCLASS command. Otherwise, storage agents might not have access to the volumes. For details, see "DEFINE PATH (Define a path)" on page 240. | /data/fpool01.dsm   |



Table 109. Volume name requirements by device type (continued)

| Device Type     | Volume Name Requirements  | Example   |
|-----------------|---|-----------|
| TAPE            | 1–32 alphanumeric characters for all device types except CARTRIDGE, the range for which is 1–6 characters | DSMT01    |
|                 | The server converts volume names to uppercase.  |           |
| OPTICAL or WORM | 1–32 alphanumeric characters  | DSM_SP001 |
|                 | The server converts volume names to uppercase.  |           |
| REMOVABLEFILE   | 1–6 alphanumeric characters   | DSM01     |
|                 | The server converts volume names to uppercase.  |           |

### ACcess

Specifies how client nodes and server processes (such as migration) can access files in the storage pool volume. This parameter is optional. The default value is READWRITE. Possible values are:

#### READWrite

Specifies that client nodes and server processes can read from and write to files stored on the volume.

#### READOnly

Specifies that client nodes and server processes can only read files stored on the volume.

#### UNAVailable

Specifies that neither client nodes nor server processes 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, the server does not attempt to access the volume.

#### Offsite

Specifies that the volume is at an offsite location from which it cannot be mounted. You can specify this value only for volumes in copy storage pools.

Use this value to help you track volumes at offsite locations. The server treats volumes designated as offsite differently:

- The server does not generate mount requests for volumes designated offsite.
- The server reclaims or moves data from offsite volumes by retrieving files from other storage pools.
- The server does not automatically delete empty, offsite scratch volumes from a copy storage pool.

### 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. Enclose the location in quotation marks if it contains any blank characters.

### FormatSize

Specifies the size of the random access volume or FILE volume that is created

## DEFINE VOLUME

and formatted in one step. The value is specified in megabytes. The maximum size is 8 000 000 MB (8 terabytes). This parameter is required if any of the following conditions are true:

- A single FILE or DISK volume is specified, which is to be created and formatted in one step.
- The value for the NUMBEROFVOLUMES parameter is greater than 1, and DISK volumes are being created.
- The value of the NUMBEROFVOLUMES parameter is greater than 1, and the value of the FORMATSIZ parameter is less than or equal to the MAXCAPACITY parameter of the DEFINE DEVCLASS command.

For a FILE volume, you must specify a value less than or equal to the value of the MAXCAPACITY parameter of the device class associated with the storage pool.

You cannot use this parameter for multiple, predefined volumes. Unless WAIT=YES is specified, the operation is performed as a background process.

### Numberofvolumes

Specifies the number of volumes that are created and formatted in one step. This parameter applies only to storage pools with DISK- or FILE-type device classes. This parameter is optional. The default is 1. If you specify a value greater than 1, you must also specify a value for the FORMATSIZ parameter. Specify a number from 1 to 256.

If the value for the NUMBEROFVOLUMES parameter is greater than 1, the volume name you specified will have a numeric suffix appended to create each name (for example, tivoli001, tivoli002, and so on). Be sure to choose a volume name so that a valid file name for the target file system is created when the suffix is appended.

**Note:** You must ensure that storage agents can access newly created FILE volumes. For more information, see “DEFINE PATH (Define a path)” on page 240.

### Wait

Specifies whether volume creation and formatting operation is performed in the foreground or background. This parameter is optional. It is ignored unless you also specify the FORMATSIZ parameter.

#### No

Specifies that a volume creation and formatting operation is performed in the background. This is the default when you also specify a format size.

#### Yes

Specifies that a volume creation and formatting operation is performed in the foreground.

**Note:** You cannot specify WAIT=YES from the server console.

### Example: Use a background process to define a new 100 MB volume for a disk storage pool

Create a new volume of 100 MB in the disk storage pool named BACKUPPOOL. The volume name is /var/storage/bf.dsm. Let the volume be created as a background process.

```
define volume backuppool  
/var/storage/bf.dsm formatsize=100
```

### Example: Define a volume to a disk storage pool with read and write access

A storage pool named POOL1 is assigned to a tape device class. Define a volume named TAPE01 to this storage pool, with READWRITE access.

```
define volume pool1 tape01 access=readwrite
```

### Example: Define a volume to a file storage pool

A storage pool named FILEPOOL is assigned to a device class with a device type of FILE. Define a volume named filepool\_vol01 to this storage pool.

```
define volume filepool /usr/storage/filepool_vol01
```

### Example: Assign a volume to an optical storage pool with read and write access

A storage pool named OPOOL is assigned to a device class with a device type of OPTICAL. Define a volume named LONGTERMVOL to this storage pool.

```
define volume opool longtermvol
```

### Example: Example: Use a background process to define 10 volumes for a file storage pool with a device class 5 GB maximum capacity

Define ten volumes in a sequential storage pool that uses a FILE device class. The storage pool is named FILEPOOL. The value of the MAXCAPACITY parameter for the device class associated with this storage pool is 5 GB. Creation will occur in the background.

```
define volume filepool filevol numberofvolumes=10 formatsize=5000
```

The Tivoli Storage Manager server creates volume names filevol001 through filevol010.

Volumes are created in the directory or directories specified with the DIRECTORY parameter of the device class associated with storage pool filepool. If you specified multiple directories for the device class, individual volumes can be created in any of the directories in the list.

## Related commands

Table 110. Commands related to DEFINE VOLUME

| Command          | Description   |
|------------------|---|
| DEFINE STGPOOL   | Defines a storage pool as a named collection of server storage media. |
| DELETE VOLUME    | Deletes a volume from a storage pool.                                 |
| QUERY VOLUME     | Displays information about storage pool volumes.                      |
| UPDATE DEVCLASS  | Changes the attributes of a device class.                             |
| UPDATE LIBVOLUME | Changes the status of a storage volume.                               |
| UPDATE VOLUME    | Updates the attributes of storage pool volumes.                       |

---

## DELETE commands

Use the DELETE commands to delete or remove a Tivoli Storage Manager object.

The following is a list of DELETE commands for Tivoli Storage Manager:

- “DELETE ASSOCIATION (Delete the node association to a schedule)” on page 344
- “DELETE BACKUPSET (Delete a backup set)” on page 346
- “DELETE CLIENTOPT (Delete an option in an option set)” on page 351
- “DELETE CLOPTSET (Delete a client option set)” on page 352
- “DELETE COLLOCGROUP (Delete a collocation group)” on page 353
- “DELETE COLLOCMEMBER (Delete collocation group member)” on page 354
- “DELETE COPYGROUP (Delete a backup or archive copy group)” on page 356
- “DELETE DATAMOVER (Delete a data mover)” on page 358
- “DELETE DEVCLASS (Delete a device class)” on page 359
- “DELETE DOMAIN (Delete a policy domain)” on page 360
- “DELETE DRIVE (Delete a drive from a library)” on page 361
- “DELETE EVENT (Delete event records)” on page 362
- “DELETE EVENTSERVER (Delete the definition of the event server)” on page 364
- “DELETE FILESPACE (Delete client node data from the server)” on page 365
- “DELETE GRPMEMBER (Delete a server from a server group)” on page 369
- “DELETE KEYRING (Delete password information in the certificate key database)” on page 370
- “DELETE LIBRARY (Delete a library)” on page 371
- “DELETE MACHINE (Delete machine information)” on page 372
- “DELETE MACHNODEASSOCIATION (Delete association between a machine and a node)” on page 373
- “DELETE MGMTCLASS (Delete a management class)” on page 374
- “DELETE NODEGROUP (Delete a node group)” on page 375
- “DELETE NODEGROUPMEMBER (Delete node group member)” on page 376
- “DELETE PATH (Delete a path)” on page 377
- “DELETE POLICYSET (Delete a policy set)” on page 379
- “DELETE PROFASSOCIATION (Delete a profile association)” on page 380
- “DELETE PROFILE (Delete a profile)” on page 383
- “DELETE RECMEDMACHASSOCIATION (Delete recovery media and machine association)” on page 385
- “DELETE RECOVERYMEDIA (Delete recovery media)” on page 386
- “DELETE SCHEDULE (Delete a client or an administrative command schedule)” on page 387
- “DELETE SCRIPT (Delete command lines from a script or delete the entire script)” on page 390
- “DELETE SERVER (Delete a server definition)” on page 391
- “DELETE SERVERGROUP (Delete a server group)” on page 392
- “DELETE SPACETRIGGER (Delete the storage pool space triggers)” on page 393
- “DELETE STGPOOL (Delete a storage pool)” on page 394

- “DELETE SUBSCRIBER (Delete subscriptions from a configuration manager database)” on page 395
- “DELETE SUBSCRIPTION (Delete a profile subscription)” on page 396
- “DELETE VIRTUALFSMAPPING (Delete a virtual file space mapping)” on page 397
- “DELETE VOLHISTORY (Delete sequential volume history information)” on page 398
- “DELETE VOLUME (Delete a storage pool volume)” on page 403

## DELETE ASSOCIATION (Delete the node association to a schedule)

Use this command to delete the association of a client node to a client schedule. Tivoli Storage Manager no longer runs the schedule on the client node.

If you try to disassociate a client from a schedule to which it is not associated, this command has no effect for that client.

### Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege
- Restricted policy privilege for the domain to which the schedule belongs

### Syntax

```

▶▶—DELEte ASSOCiation—domain_name—schedule_name—node_name—▶▶

```

### Parameters

#### *domain\_name* (Required)

Specifies the name of the policy domain to which the schedule belongs.

#### *schedule\_name* (Required)

Specifies the name of the schedule from which clients are to be disassociated.

#### *node\_name* (Required)

Specifies the name of the client node that is no longer associated with the client schedule. You can specify a list of clients which are to be no longer associated with the specified schedule. Commas, with no intervening spaces, separate the items in the list. You can also use a wildcard character to specify a name. All matching clients are disassociated from the specified schedule.

### Example: Delete a node association to a schedule

To delete the association of the node JEFF, assigned to the DOMAIN1 policy domain, to the WEEKLY\_BACKUP schedule issue the following command:

```
delete association domain1 weekly_backup jeff
```

### Example: Delete a node association to a schedule using a wildcard for node selection

Delete the association of selected clients, assigned to the DOMAIN1 policy domain, to the WEEKLY\_BACKUP schedule so that this schedule is no longer run by these clients. The nodes that are disassociated from the schedule contain ABC or XYZ in the node name. Issue the command:

```
delete association domain1 weekly_backup *abc*,*xyz*
```

**Related commands***Table 111. Commands related to DELETE ASSOCIATION*

| Command            | Description   |
|--------------------|---|
| DEFINE ASSOCIATION | Associates clients with a schedule.                         |
| QUERY ASSOCIATION  | Displays the clients associated with one or more schedules. |

## DELETE BACKUPSET

### DELETE BACKUPSET (Delete a backup set)

Use this command to manually delete a backup set before its retention period expires.

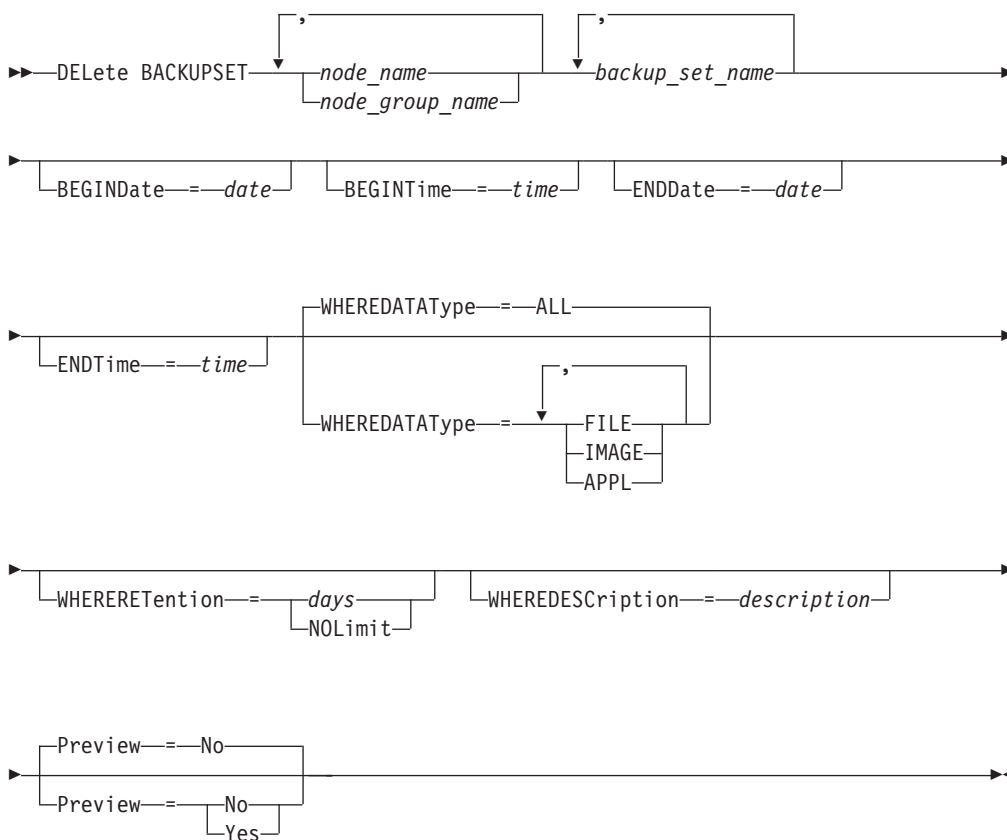
When the server creates a backup set, the retention period assigned to the backup set determines how long the backup set remains in the database. When that date passes, the server automatically deletes the backup set when expiration processing runs. However, you can also manually delete the client's backup set from the server before it is scheduled to expire by using the DELETE BACKUPSET command.

**Attention:** If the volumes contain multiple backup sets, they are not returned to scratch status until all the backup sets are expired or are deleted.

#### Privilege class

If the REQSYSAUTHOUTFILE server option is set to YES (the default), the administrator must have system privilege. If the REQSYSAUTHOUTFILE server option is set to NO, the administrator must have system privilege or policy privilege for the domain to which the client node is assigned.

#### Syntax





## Parameters

### *node\_name* or *node\_group\_name* (Required)

Specifies the name of the client nodes or node groups whose data is contained in the specified backup set volumes. To specify multiple node and node group names, separate the names with commas and no intervening spaces. Any node name you specify may contain wildcard characters, but node group names cannot contain wildcard characters. If backup set volumes contain backup sets from multiple nodes then every backup set whose node name matches one of the specified node names will be deleted.

### *backup\_set\_name* (Required)

Specifies the name of the backup set to delete. The backup set name you specify can contain wildcard characters. You can specify more than one backup set name by separating the names with commas and no intervening spaces.

### BEGINDate

Specifies the beginning date in which the backup set to delete was created. This parameter is optional. You can use this parameter with the **BEGINTIME** parameter to specify a range for the date and time. If you specify a begin date without a begin time, the time will be at 12:00 a.m. (midnight) on the date you specify.

You can specify the date by using one of the following values:

| Value                                  | Description                            | Example         |
|--|--|-----------------|
| <i>MM/DD/YYYY</i>                      | A specific date                        | 09/15/1999      |
| TODAY                                  | The current date                       | TODAY           |
| TODAY+ <i>days</i> or<br>+ <i>days</i> | The current date plus days specified.  | TODAY +3 or +3. |
| TODAY- <i>days</i> or<br>- <i>days</i> | The current date minus days specified. | TODAY -3 or -3. |

### BEGINTime

Specifies the beginning time in which the backup set to delete was created. This parameter is optional. You can use this parameter in conjunction with the **BEGINDATE** parameter to specify a range for the date and time. If you specify a begin time without a begin date, the date will be the current date at the time you specify.

You can specify the time by using one of the following values:

| Value                                  | Description  | Example              |
|--|--|----------------------|
| <i>HH:MM:SS</i>                        | A specific time                                    | 10:30:08             |
| NOW                                    | The current time                                   | NOW                  |
| NOW+ <i>HH:MM</i> or<br>+ <i>HH:MM</i> | The current time plus hours and minutes specified  | NOW+02:00 or +02:00. |
| NOW- <i>HH:MM</i> or<br>- <i>HH:MM</i> | The current time minus hours and minutes specified | NOW-02:00 or -02:00. |

### ENDDate

Specifies the ending date in which the backup set to delete was created. This parameter is optional. You can use this parameter in conjunction with the **ENDTIME** parameter to specify a range for the date and time. If you specify an end date without an end time, the time will be at 11:59:59 p.m. on the specified end date.

## DELETE BACKUPSET

You can specify the date by using one of the following values:

| Value                                  | Description                            | Example         |
|--|--|-----------------|
| <i>MM/DD/YYYY</i>                      | A specific date                        | 09/15/1999      |
| TODAY                                  | The current date                       | TODAY           |
| TODAY+ <i>days</i> or<br>+ <i>days</i> | The current date plus days specified.  | TODAY +3 or +3. |
| TODAY- <i>days</i> or<br>- <i>days</i> | The current date minus days specified. | TODAY -3 or -3. |

### ENDTime

Specifies the ending time of the range in which the backup set to delete was created. This parameter is optional. You can use this parameter in conjunction with the ENDDATE parameter to specify a range for the date and time. If you specify an end time without an end date, the date will be the current date at the time you specify.

You can specify the time by using one of the following values:

| Value                                  | Description  | Example              |
|--|--|----------------------|
| <i>HH:MM:SS</i>                        | A specific time  | 10:30:08             |
| NOW                                    | The current time   | NOW                  |
| NOW+ <i>HH:MM</i> or<br>+ <i>HH:MM</i> | The current time plus hours and minutes on the specified end date  | NOW+02:00 or +02:00. |
| NOW- <i>HH:MM</i> or<br>- <i>HH:MM</i> | The current time minus hours and minutes on the specified end date | NOW-02:00 or -02:00. |

### WHERE DATATYPE

Specifies the backup sets containing the specified types of data are to be deleted. This parameter is optional. The default is that backup sets for all types of data (file level, image, and application) are to be deleted. To specify multiple data types, separate the data types with commas and no intervening spaces. Possible values are:

#### ALL

Specifies that backup sets for all types of data (file level, image, and application) are to be deleted. This is the default.

#### FILE

Specifies that a file level backup set is to be deleted. File level backup sets contain files and directories backup up by the backup-archive client.

#### IMAGE

Specifies that an image backup set is to be deleted. Image backup sets contain images created by the backup-archive client BACKUP IMAGE command.

#### APPL

Specifies that an application backup set is to be deleted. Application backup sets contain data backed up to the Tivoli Storage Manager server by applications using the Tivoli Storage Manager application programming interface. This parameter applies only if the Tivoli Storage Manager server

was upgraded from Tivoli Storage Manager Express and only for the Tivoli Storage Manager Express versions of Data Protection for Exchange and Data Protection for SQL.

#### WHERERetention

Specifies the retention value, specified in days, that is associated with the backup sets to delete. You can specify an integer from 0 to 30000. The values are:

*days*

Specifies that backup sets that are retained this number of days are deleted.

#### NOLimit

Specifies that the backup sets that are retained indefinitely are deleted.

#### WHEREDESCRIPTION

Specifies the description that is associated with the backup set to delete. The description you specify can contain a wildcard character. This parameter is optional. Enclose the description in quotation marks if it contains any blank characters.

#### Preview

Specifies whether to preview the list of backup sets to delete, without actually deleting the backup sets. This parameter is optional. The default value is NO. The values are:

#### No

Specifies that the backup sets are deleted.

#### Yes

Specifies that the server displays the list of backup sets to delete, without actually deleting the backup sets.

### Example: Delete a backup set

Delete backup set named PERS\_DATA.3099 that belongs to client node JANE. The backup set was generated on 11/19/1998 at 10:30:05 and the description is "Documentation Shop".

```
delete backupset pers_data.3099
begindate=11/19/1998 begintime=10:30:05
wheredescription="documentation shop"
```

### Related commands

Table 112. Commands related to DELETE BACKUPSET

| Command                | Description  |
|------------------------|--|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server. |
| DEFINE NODEGROUP       | Defines a group of nodes.                              |
| DEFINE NODEGROUPMEMBER | Adds a client node to a node group.                    |
| DELETE NODEGROUP       | Deletes a node group.                                  |
| DELETE NODEGROUPMEMBER | Deletes a client node from a node group.               |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.             |
| GENERATE BACKUPSETTOC  | Generates a table of contents for a backup set.        |
| QUERY BACKUPSET        | Displays backup sets.                                  |

## DELETE BACKUPSET

*Table 112. Commands related to DELETE BACKUPSET (continued)*

| Command                 | Description   |
|-------------------------|---|
| QUERY NODEGROUP         | Displays information about node groups.                 |
| QUERY BACKUPSETCONTENTS | Displays contents contained in backup sets.             |
| UPDATE BACKUPSET        | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP        | Updates the description of a node group.                |



## DELETE CLOPTSET (Delete a client option set)

Use this command to delete a client option set.

### Privilege class

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

### Syntax

►►—DELEte CLOptset—*option\_set\_name*—◄◄

### Parameters

*option\_set\_name* **(Required)**

Specifies the name of the client option set to delete.

### Example: Delete a client option set

Delete the client option set named ENG.

```
delete cloptset eng
```

### Related commands

Table 114. Commands related to DELETE CLOPTSET

| Command          | Description  |
|------------------|--|
| COPY CLOPTSET    | Copies a client option set.  |
| DEFINE CLIENTOPT | Adds a client option to a client option set.                           |
| DEFINE CLOPTSET  | Defines a client option set.   |
| DELETE CLIENTOPT | Deletes a client option from a client option set.                      |
| QUERY CLOPTSET   | Displays information about a client option set.                        |
| UPDATE CLIENTOPT | Updates the sequence number of a client option in a client option set. |
| UPDATE CLOPTSET  | Updates the description of a client option set.                        |

## DELETE COLLOCGROUP (Delete a collocation group)

Use this command to delete a collocation group. You cannot delete a collocation group if it has any members in it.

You can remove all the members in the collocation group by issuing the DELETE COLLOCMEMBER command with a wildcard in the *node\_name* parameter.

### Privilege class

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

### Syntax

```
►►—DELEte COLLOCGroup—group_name—◄◄
```

### Parameters

*group\_name*

Specifies the name of the collocation group that you want to delete.

### Example: Delete a collocation group

Delete a collocation group named group1.

```
delete collocgroup group1
```

### Related commands

Table 115. Commands related to DELETE COLLOCGROUP

| Command             | Description  |
|---------------------|--|
| DEFINE COLLOCGROUP  | Defines a collocation group.   |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                       |
| DEFINE STGPOOL      | Defines a storage pool as a named collection of server storage media.            |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                  |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces.    |
| QUERY COLLOCGROUP   | Displays information about collocation groups.                                   |
| QUERY NODE          | Displays partial or complete information about one or more clients.              |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.      |
| QUERY STGPOOL       | Displays information about storage pools.  |
| REMOVE NODE         | Removes a client from the list of registered nodes for a specific policy domain. |
| UPDATE COLLOCGROUP  | Updates the description of a collocation group.                                  |
| UPDATE STGPOOL      | Changes the attributes of a storage pool.  |

## DELETE COLLOCMEMBER (Delete collocation group member)

Use this command to delete a client node from a collocation group.

### Privilege class

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

### Syntax

```

DELEte COLLOCMember—group_name—node_name

```

### Parameters

*group\_name*

Specifies the name of the collocation group from which you want to delete a client node.

*node\_name*

Specifies the name of the client node that you want to delete from the collocation group. You can specify one or more names. When specifying multiple names, separate the names with commas; do not use intervening spaces. You can also use wildcard characters to specify multiple nodes.

### Example: Delete collocation group member

Delete two nodes, NODE1 and NODE2, from a collocation group, GROUP1.

```
delete collocmember group1 node1,node2
```

### Related commands

Table 116. Commands related to DELETE COLLOCMEMBER

| Command             | Description  |
|---------------------|--|
| DEFINE COLLOGROUP   | Defines a collocation group.   |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                       |
| DEFINE STGPOOL      | Defines a storage pool as a named collection of server storage media.            |
| DELETE COLLOGROUP   | Deletes a collocation group.   |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces.    |
| QUERY COLLOGROUP    | Displays information about collocation groups.                                   |
| QUERY NODE          | Displays partial or complete information about one or more clients.              |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.      |
| QUERY STGPOOL       | Displays information about storage pools.  |
| REMOVE NODE         | Removes a client from the list of registered nodes for a specific policy domain. |
| UPDATE COLLOGROUP   | Updates the description of a collocation group.                                  |



*Table 116. Commands related to DELETE COLLOCMEMBER (continued)*

| Command        | Description                               |
|----------------|---|
| UPDATE STGPOOL | Changes the attributes of a storage pool. |

## DELETE COPYGROUP (Delete a backup or archive copy group)

Use this command to delete a backup or archive copy group from a management class. You cannot delete a copy group in the ACTIVE policy set.

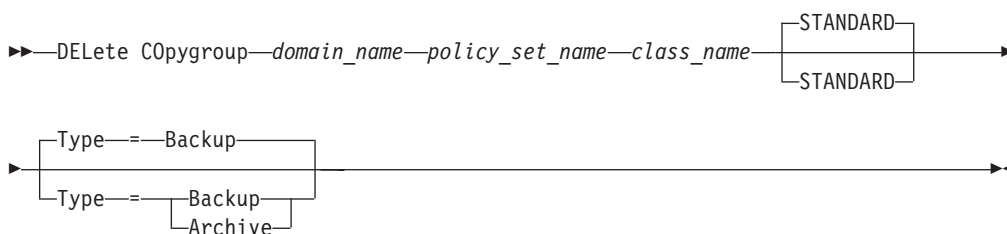
When you activate the changed policy set, any files that are bound to a deleted copy group are managed by the default management class.

You can delete the predefined STANDARD copy group in the STANDARD policy domain (STANDARD policy set, STANDARD management class). However, if you later reinstall the Tivoli Storage Manager server, the process restores all STANDARD policy objects.

### 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

#### *domain\_name* (Required)

Specifies the policy domain to which the copy group belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set to which the copy group belongs.

#### *class\_name* (Required)

Specifies the management class to which the copy group belongs.

#### STANDARD

Specifies the copy group, which is always **STANDARD**. This parameter is optional. The default value is **STANDARD**.

#### Type

Specifies the type of copy group to delete. This parameter is optional. The default value is **BACKUP**. Possible values are:

##### **Backup**

Specifies that the backup copy group is deleted.

##### **Archive**

Specifies that the archive copy group is deleted.

**Example: Delete a backup copy group**

Delete the backup copy group from the ACTIVEFILES management class that is in the VACATION policy set of the EMPLOYEE\_RECORDS policy domain.

```
delete copygroup employee_records
vacation activefiles
```

**Example: Delete an archive copy group**

Delete the archive copy group from the MCLASS1 management class that is in the SUMMER policy set of the PROG1 policy domain.

```
delete copygroup prog1 summer mclass1 type=archive
```

**Related commands**

*Table 117. Commands related to DELETE COPYGROUP*

| Command          | Description  |
|------------------|--|
| DEFINE COPYGROUP | Defines a copy group for backup or archive processing within a specified management class. |
| QUERY COPYGROUP  | Displays the attributes of a copy group.   |
| UPDATE COPYGROUP | Changes one or more attributes of a copy group.  |

## DELETE DATAMOVER

### DELETE DATAMOVER (Delete a data mover)

Use this command to delete a data mover. You cannot delete the data mover if any paths are defined for this data mover.

#### Privilege class

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

#### Syntax

►►—DELEte DATAMover—*data\_mover\_name*—◄◄

#### Parameters

*data\_mover\_name* **(Required)**

Specifies the name of the data mover.

**Note:** This command deletes the data mover even if there is data for the corresponding NAS node.

#### Example: Delete a data mover

Delete the data mover for the node named NAS1.

```
delete datamover nas1
```

#### Related commands

Table 118. Commands related to DELETE DATAMOVER

| Command          | Description   |
|------------------|---|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server.      |
| DEFINE PATH      | Defines a path from a source to a destination.                      |
| DELETE PATH      | Deletes a path from a source to a destination.                      |
| QUERY DATAMOVER  | Displays data mover definitions.                                    |
| QUERY PATH       | Displays information about the path from a source to a destination. |
| UPDATE DATAMOVER | Changes the definition for a data mover.                            |

## DELETE DEVCLASS (Delete a device class)

Use this command to delete a device class.

To use this command, you must first delete all storage pools that are assigned to the device class and, if necessary, cancel any database export or import processes that are using the device class.

You cannot delete the device class DISK, which is predefined at installation, but you can delete any device classes defined by the Tivoli Storage Manager administrator.

### Privilege class

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

### Syntax

►►—DELEte DEVclass—*device\_class\_name*—◄◄

### Parameters

*device\_class\_name* **(Required)**

Specifies the name of the device class to be deleted.

### Example: Delete a device class

Delete the device class named MYTAPE. There are no storage pools assigned to the device class.

```
delete devclass mytape
```

### Related commands

Table 119. Commands related to DELETE DEVCLASS

| Command         | Description                                  |
|-----------------|--|
| DEFINE DEVCLASS | Defines a device class.                      |
| QUERY DEVCLASS  | Displays information about device classes.   |
| QUERY DIRSPACE  | Displays information about FILE directories. |
| UPDATE DEVCLASS | Changes the attributes of a device class.    |

### DELETE DOMAIN (Delete a policy domain)

Use this command to delete a policy domain. All associated policy sets, management classes, and copy groups are deleted along with the policy domain. You cannot delete a policy domain to which client nodes are registered.

You can delete the predefined STANDARD policy domain. However, if you later reinstall the Tivoli Storage Manager server, the process restores all STANDARD policy objects.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

►►—DELEte D0main—*domain\_name*—◄◄

#### Parameters

*domain\_name* (**Required**)

Specifies the policy domain to delete.

#### Examples: Delete a policy domain

Delete the EMPLOYEE\_RECORDS policy domain.

```
delete domain employee_records
```

#### Related commands

Table 120. Commands related to DELETE DOMAIN

| Command       | Description  |
|---------------|--|
| COPY DOMAIN   | Creates a copy of a policy domain.                       |
| DEFINE DOMAIN | Defines a policy domain that clients can be assigned to. |
| QUERY DOMAIN  | Displays information about policy domains.               |
| UPDATE DOMAIN | Changes the attributes of a policy domain.               |

## DELETE DRIVE (Delete a drive from a library)

Use this command to delete a drive from a library. A drive that is in use cannot be deleted.

All paths related to a drive must be deleted before the drive itself can be deleted.

### Privilege class

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

### Syntax

►►—DELEte DRive—*library\_name*—*drive\_name*—◄◄

### Parameters

#### *library\_name* (Required)

Specifies the name of the library where the drive is located.

#### *drive\_name* (Required)

Specifies the name of the drive to be deleted.

### Example: Delete a drive from a library

Delete DRIVE3 from the library named AUTO.

```
delete drive auto drive3
```

### Related commands

Table 121. Commands related to DELETE DRIVE

| Command        | Description                                       |
|----------------|---|
| DEFINE DRIVE   | Assigns a drive to a library.                     |
| DEFINE LIBRARY | Defines an automated or manual library.           |
| DELETE LIBRARY | Deletes a library.                                |
| DELETE PATH    | Deletes a path from a source to a destination.    |
| QUERY DRIVE    | Displays information about drives.                |
| QUERY LIBRARY  | Displays information about one or more libraries. |
| UPDATE DRIVE   | Changes the attributes of a drive.                |

## DELETE EVENT (Delete event records)

Use this command to delete event records from the database. An event record is created whenever processing of a scheduled command is started or missed.

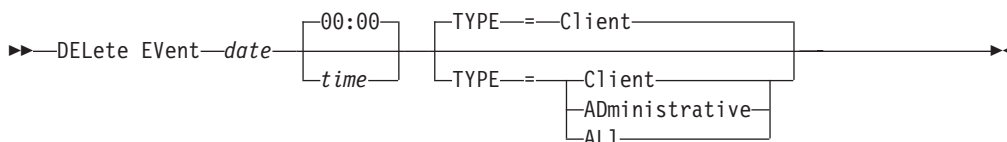
This command only deletes the event records that exist at the time the command is processed. An event record will not be found:

- If the event record has never been created (the event is scheduled for the future)
- If the event has passed and the event record has already been deleted.

### Privilege class

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

### Syntax



### Parameters

#### *date* (Required)

Specifies the date used to determine which event records to delete. The maximum number of days you can specify is 9999.

Use this parameter in conjunction with the TIME parameter to specify a date and time for deleting event records. Any record whose scheduled start occurs before the specified date and time is deleted. However, records are not deleted for events whose startup window has not yet passed.

You can specify the date by using one of the following values:

| Value               | Description                           | Example        |
|---------------------|---------------------------------------|----------------|
| MM/DD/YYYY          | A specific date                       | 09/15/1998     |
| TODAY               | The current date                      | TODAY          |
| TODAY-days or -days | The current date minus days specified | TODAY-3 or -3. |

#### *time*

Specifies the time used to determine which event records to delete. Use this parameter in conjunction with the DATE parameter to specify a date and time for deleting event records. Any record whose scheduled start occurs before the specified date and time is deleted. However, records are not deleted for events whose startup window has not yet passed. The default is 00:00.

You can specify the time by using one of the following values:

| Value    | Description      | Example  |
|----------|------------------|----------|
| HH:MM:SS | A specific time  | 10:30:08 |
| NOW      | The current time | NOW      |



| Value                  | Description  | Example   |
|------------------------|--|---|
| NOW+HH:MM<br>or +HH:MM | The current time plus hours and minutes specified  | NOW+03:00 or +03:00<br><br><b>Attention:</b> If you issue this command at 9:00 using NOW+03:00 or +03:00, Tivoli Storage Manager deletes records with a time of 12:00 or later on the date you specify. |
| NOW-HH:MM<br>or -HH:MM | The current time minus hours and minutes specified | NOW-03:00 or -03:00   |

**TYPE**

Specifies the type of events to be deleted. This parameter is optional. The default is CLIENT. Possible values are:

**Client**

Specifies to delete event records for client schedules.

**Administrative**

Specifies to delete event records for administrative command schedules.

**ALL**

Specifies to delete event records for both client and administrative command schedules.

**Example: Delete event records**

Delete records for events with scheduled start times prior to 08:00 on May 26, 1998 (05/26/1998), 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.

```
delete event 05/26/1998 08:00
```

**Related commands**

Table 122. Commands related to DELETE EVENT

| Command            | Description   |
|--------------------|---|
| QUERY EVENT        | Displays information about scheduled and completed events for selected clients. |
| SET EVENTRETENTION | Specifies the number of days to retain records for scheduled operations.        |

### DELETE EVENTSERVER (Delete the definition of the event server)

Use this command to delete the definition of the event server. You must issue this command before you issue the DELETE SERVER command. If you specify the server defined as the event server on the DELETE SERVER command, you will receive an error message.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

►►—DELeTe EVENTSErVer—————►◄

#### Example: Delete an event server definition

Delete the definition for the event server ASTRO.

```
delete eventserver
```

#### Related commands

*Table 123. Commands related to DELETE EVENTSERVER*

| Command            | Description                            |
|--------------------|--|
| DEFINE EVENTSERVER | Defines a server as an event server.   |
| QUERY EVENTSERVER  | Displays the name of the event server. |

## DELETE FILESPACE (Delete client node data from the server)

Use this command to delete file spaces from the server. Files that belong to the file space are deleted from primary, active-data, and copy storage pools.

Tivoli Storage Manager 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 process is canceled or if a system failure occurs, a partial deletion can occur. A subsequent DELETE FILESPACE command for the same node or owner can delete the remaining data.

If this command is applied to a WORM (write once, read many) volume, the volume will return to scratch if it has space remaining in which data can be written. (Note that data on WORM volumes, including deleted and expired data, cannot be overwritten. Therefore, data can only be written in space that does not contain current, deleted, or expired data.) If a WORM volume does not have any space available in which data can be written, it will remain private. To remove the volume from the library, you must use the CHECKOUT LIBVOLUME command.

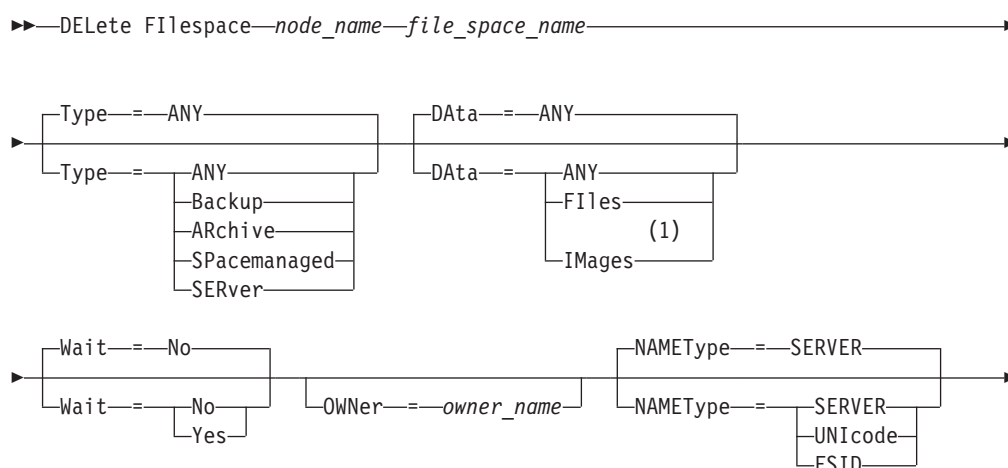
### Important:

- If archive retention protection is enabled, the server will only delete archive files with expired retention periods. See the SET ARCHIVERETENTIONPROTECTION command for more information.
- The server will not delete archive files that are on deletion hold until the hold is released.
- Reclamation will not start while the DELETE FILESPACE process is running.

### 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



### Notes:

- 1 This parameter can only be used when TYPE=ANY or TYPE=BACKUP is specified.

## Parameters

### *node\_name* (Required)

Specifies the name of the client node to which the file space belongs.

### *file\_space\_name* (Required)

Specifies the name of the file space to be deleted. This name is case-sensitive and must be entered exactly as it is known to the server. To determine how to enter the name, use the QUERY FILESPACE command. You can use wildcard characters to specify this name.

For a server that has clients with support for Unicode, you may need to have the server convert the file space name that you enter. For example, you may need to have the server convert the name you enter from the server's code page to Unicode. See the **NAMETYPE** parameter for details. If you do not specify a file space name, or specify only a single wildcard character for the name, you can use the **CODETYPE** parameter to limit the operation to Unicode file spaces or to non-Unicode file spaces.

### Type

Specifies the type of data to be deleted. This parameter is optional. The default value is ANY. Possible values are:

#### **ANY**

Delete only backed-up versions of files and archived copies of files.

If you specify delete filesystem *node\_name* \* type=any, all backed-up data and archived data in all file spaces for that node are deleted. File spaces are deleted only if they do not contain files that are migrated from a Tivoli Storage Manager for Space Management client.

#### **Backup**

Delete backup data for the file space.

#### **ARchive**

Delete all archived data on the server for the file space.

#### **SPacemanaged**

Delete files migrated from a user's local file system by a Tivoli Storage Manager for Space Management client. The **OWNER** parameter is ignored when you specify TYPE=SPACEMANAGED.

#### **SERver**

Delete all archived files in all file spaces for a node that is registered as TYPE=SERVER.

### Data

Specifies objects to delete. This parameter is optional. The default value is ANY. Possible values are:

**ANY**

Delete files, directories, and images.

**Files**

Delete files and directories.

**IMages**

Delete image objects. You can only use this parameter if you have specified TYPE=ANY or TYPE=BACKUP.

**Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default value is No. Possible values are:

**No**

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

**Yes**

Specifies that the server processes this command in the foreground. Wait for the command to complete before continuing with other tasks. The server displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

**OWNer**

Restricts the data that is deleted to files belonging to the owner. This parameter is optional; it is ignored when TYPE=SPACEMANAGED. This parameter only applies to multiuser client systems such as AIX, Linux, and Sun OS.

**NAMEType**

Specify how you want the server to interpret the file space names that you enter. This parameter is useful when the server has clients with support for Unicode. A backup-archive client with support for Unicode is currently available only for the following operating systems: Windows, Macintosh OS X, and NetWare operating systems.

Use this parameter only when you enter a partly or fully qualified file space name. The default value is SERVER. Possible values are:

**SERVER**

The server uses the server's code page to interpret the file space names.

**UNICODE**

The server converts the file space names from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the name and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

**FSID**

The server interprets the file space names as their file space IDs (FSIDs).

**CODEType**

Specify what type of file spaces are to be included in the operation. The default

## DELETE FILESPACE

is BOTH, meaning that file spaces are included regardless of code page type. Use this parameter only when you enter a single wildcard character for the file space name. Possible values are:

### UNICODE

Include only file spaces that are in Unicode.

### NONUNICODE

Include only file spaces that are not in Unicode.

### BOTH

Include file spaces regardless of code page type.

### Example: Delete a file space

Delete the file space named C\_Drive that belongs to the client node HTANG.

```
delete filesystem htang C_Drive
```

### Example: Delete all space-managed files for a client node

Delete all files migrated from client node APOLLO (that is, all space-managed files).

```
delete filesystem apollo * type=spacemanaged
```

## Related commands

Table 124. Commands related to DELETE FILESPACE

| Command          | Description  |
|------------------|--|
| CANCEL PROCESS   | Cancels a background server process.   |
| QUERY ACTLOG     | Displays messages from the server activity log.                                  |
| QUERY FILESPACE  | Displays information about data in file spaces that belong to a client.          |
| QUERY OCCUPANCY  | Displays file space information by storage pool.                                 |
| QUERY PROCESS    | Displays information about background processes.                                 |
| REMOVE NODE      | Removes a client from the list of registered nodes for a specific policy domain. |
| RENAME FILESPACE | Renames a client filesystem on the server.                                       |

## DELETE GRPMEMBER (Delete a server from a server group)

Use this command to delete a server or server group from a server group.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>>—DELEte GRPMEMber—group_name—member_name—

```

### Parameters

*group\_name* **(Required)**

Specifies the group.

*member\_name* **(Required)**

Specifies the server or group to delete from the group. To specify multiple names, separate the names with commas and no intervening spaces.

### Example: Delete a server from a server group

Delete member PHOENIX from group WEST\_COMPLEX.

```
delete grpmember west_complex phoenix
```

### Related commands

Table 125. Commands related to DELETE GRPMEMBER

| Command            | Description                                     |
|--------------------|---|
| DEFINE GRPMEMBER   | Defines a server as a member of a server group. |
| DEFINE SERVERGROUP | Defines a new server group.                     |
| DELETE SERVER      | Deletes the definition of a server.             |
| DELETE SERVERGROUP | Deletes a server group.                         |
| MOVE GRPMEMBER     | Moves a server group member.                    |
| QUERY SERVER       | Displays information about servers.             |
| QUERY SERVERGROUP  | Displays information about server groups.       |
| RENAME SERVERGROUP | Renames a server group.                         |
| UPDATE SERVERGROUP | Updates a server group.                         |

### DELETE KEYRING (Delete password information in the certificate key database)

Use this command to delete the password information in the Tivoli Storage Manager database for the certificate key database file (cert.kdb).

This command is needed when the SSLTCPPOINT or SSLTCPADMINPORT options are in use and the cert.kdb file has been lost or is not recoverable. If cert.kdb does not exist and there is no entry in the database for its password, Tivoli Storage Manager automatically generates a new self-signed certificate in a replacement cert.kdb file at server startup. The administrator then distributes the new public key (that is, the corresponding cert.arm file) to the clients that are using Secure Sockets Layer (SSL).

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

►►—DELEte KEYRing—————►►

#### Parameters

None

#### Example: Delete password information in the certificate key database

The Tivoli Storage Manager administrator has deleted the current cert.kdb file and wants Tivoli Storage Manager to generate a new one at server startup for use by SSL.

```
delete keyring
```

#### Related commands

None



## DELETE LIBRARY (Delete a library)

Use this command to delete a library. Before you delete a library, you must delete other associated objects, such as the path.

Use this command to delete a library. Before you delete a library, delete the path and all associated drives.

### Privilege class

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

### Syntax

►►—DELEte LIBRary—*library\_name*—◄◄

### Parameters

#### *library\_name* (Required)

Specifies the name of the library to be deleted.

### Example: Delete a manual library

Delete the manual library named LIBR1.

```
delete library libr1
```

### Related commands

Table 126. Commands related to DELETE LIBRARY

| Command        | Description   |
|----------------|---|
| DEFINE DRIVE   | Assigns a drive to a library.                                       |
| DEFINE LIBRARY | Defines an automated or manual library.                             |
| DEFINE PATH    | Defines a path from a source to a destination.                      |
| DELETE DRIVE   | Deletes a drive from a library.                                     |
| DELETE PATH    | Deletes a path from a source to a destination.                      |
| QUERY DRIVE    | Displays information about drives.                                  |
| QUERY LIBRARY  | Displays information about one or more libraries.                   |
| QUERY PATH     | Displays information about the path from a source to a destination. |
| UPDATE DRIVE   | Changes the attributes of a drive.                                  |
| UPDATE LIBRARY | Changes the attributes of a library.                                |
| UPDATE PATH    | Changes the attributes associated with a path.                      |

## DELETE MACHINE

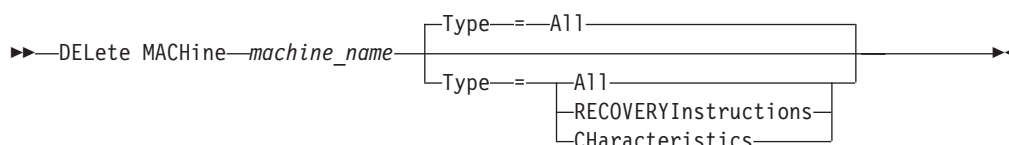
### DELETE MACHINE (Delete machine information)

Use this command to delete machine description information. To replace existing information, issue this command and then issue an INSERT MACHINE command.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax



#### Parameters

##### *machine\_name* (Required)

Specifies the name of the machine whose information is to be deleted.

##### Type

Specifies the type of machine information. This parameter is optional. The default is ALL. Possible values are:

##### All

Specifies all information.

##### RECOVERYInstructions

Specifies the recovery instructions.

##### CHaracteristics

Specifies the machine characteristics.

#### Example: Delete a specific machine's information

Delete the machine characteristics associated with the DISTRICT5 machine.

```
delete machine district5 type=characteristics
```

#### Related commands

Table 127. Commands related to DELETE MACHINE

| Command             | Description  |
|---------------------|--|
| DEFINE MACHINE      | Defines a machine for DRM.   |
| INSERT MACHINE      | Inserts machine characteristics or recovery instructions into the IBM Tivoli Storage Manager database. |
| QUERY MACHINE       | Displays information about machines.   |
| QUERY RECOVERYMEDIA | Displays media available for machine recovery.   |
| UPDATE MACHINE      | Changes the information for a machine.   |

## DELETE MACHNODEASSOCIATION (Delete association between a machine and a node)

Use this command to delete the association between a machine and one or more nodes. This command does not delete the node from Tivoli Storage Manager.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶—DELEte MACHNODEAssociation—machine_name—node_name—▶▶

```

### Parameters

#### *machine\_name* (Required)

Specifies the name of a machine that is associated with one or more nodes.

#### *node\_name* (Required)

Specifies the name of a node associated with a machine. If you specify a list of node names, separate the names with commas and no intervening spaces. You can use wildcard characters to specify a name. If a node is not associated with the machine, that node is ignored.

### Example: Delete an association between a node and a machine

Delete the association between the DISTRICT5 machine and the ACCOUNTSPAYABLE node.

```
delete machnodeassociation district5 accountspayable
```

### Related commands

Table 128. Commands related to DELETE MACHNODEASSOCIATION

| Command                    | Description   |
|----------------------------|---|
| DEFINE MACHNODEASSOCIATION | Associates an IBM Tivoli Storage Manager node with a machine. |
| QUERY MACHINE              | Displays information about machines.                          |

## DELETE MGMTCLASS (Delete a management class)

Use this command to delete a management class. You cannot delete a management class in the ACTIVE policy set. All copy groups in the management class are deleted along with the management class.

You can delete the management class assigned as the default for a policy set, but a policy set cannot be activated unless it has a default management class.

You can delete the predefined STANDARD management class in the STANDARD policy domain. However, if you later reinstall the Tivoli Storage Manager server, the process restores all STANDARD policy objects.

### 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—domain_name—policy_set_name—class_name————►►
```

### Parameters

*domain\_name* **(Required)**

Specifies the policy domain to which the management class belongs.

*policy\_set\_name* **(Required)**

Specifies the policy set to which the management class belongs.

*class\_name* **(Required)**

Specifies the management class to delete.

### Example: Delete a management class

Delete the ACTIVEFILES management class from the VACATION policy set of the EMPLOYEE\_RECORDS policy domain.

```
delete mgmtclass employee_records
vacation activefiles
```

### Related commands

*Table 129. Commands related to DELETE MGMTCLASS*

| Command             | Description   |
|---------------------|---|
| ASSIGN DEFMGMTCLASS | Assigns a management class as the default for a specified policy set. |
| COPY MGMTCLASS      | Creates a copy of a management class.                                 |
| DEFINE MGMTCLASS    | Defines a management class.   |
| QUERY MGMTCLASS     | Displays information about management classes.                        |
| UPDATE MGMTCLASS    | Changes the attributes of a management class.                         |

## DELETE NODEGROUP (Delete a node group)

Use this command to delete a node group. You cannot delete a node group if it has any members in it.

**Attention:** You can remove all the members in the node group by issuing the DELETE NODEGROUPMEMBER command with a wildcard in the *node\_name* parameter.

### Privilege class

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

### Syntax

```
►►—DELEte NODEGroup—group_name—◄◄
```

### Parameters

*group\_name*

Specifies the name of the node group that you want to delete.

### Example: Delete a node group

Delete a node group named group1.

```
delete nodegroup group1
```

### Related commands

Table 130. Commands related to DELETE NODEGROUP

| Command                | Description   |
|------------------------|---|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP       | Defines a group of nodes.                               |
| DEFINE NODEGROUPMEMBER | Adds a client node to a node group.                     |
| DELETE BACKUPSET       | Deletes a backup set.                                   |
| DELETE NODEGROUPMEMBER | Deletes a client node from a node group.                |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.              |
| QUERY BACKUPSET        | Displays backup sets.                                   |
| QUERY NODEGROUP        | Displays information about node groups.                 |
| UPDATE BACKUPSET       | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP       | Updates the description of a node group.                |

**DELETE NODEGROUPMEMBER (Delete node group member)**

Use this command to delete a client node from a node group.

**Privilege class**

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

**Syntax**

```

▶▶—DELEte NODEGROUPMember—group_name—node_name—▶▶

```

**Parameters**

*group\_name*

Specifies the name of the node group from which you want to delete a client node.

*node\_name*

Specifies the name of the client node that you want to delete from the node group. You can specify one or more names. When specifying multiple names, separate the names with commas; do not use intervening spaces. You can also use wildcard characters to specify multiple nodes.

**Example: Delete node group members**

Delete two nodes, node1 and node2, from a node group, group1.

```
delete nodegroupmember group1 node1,node2
```

**Related commands**

*Table 131. Commands related to DELETE NODEGROUPMEMBER*

| Command                | Description   |
|------------------------|---|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP       | Defines a group of nodes.                               |
| DEFINE NODEGROUPMEMBER | Adds a client node to a node group.                     |
| DELETE BACKUPSET       | Deletes a backup set.                                   |
| DELETE NODEGROUP       | Deletes a node group.                                   |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.              |
| QUERY BACKUPSET        | Displays backup sets.                                   |
| QUERY NODEGROUP        | Displays information about node groups.                 |
| UPDATE BACKUPSET       | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP       | Updates the description of a node group.                |

## DELETE PATH (Delete a path)

Use this command to delete a path definition

### Privilege class

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

### Syntax

```

▶▶—DELEte PATH—source_name—destination_name—SRCType—=—DATAMover  
SERVer—▶
└─┬─DRive—LIBRARY—=library_name—▶
└─LIBRARY—▶

```

### Parameters

#### *source\_name* (Required)

Specifies the name of the source of the path to be deleted. This parameter is required.

The name specified must be that of a server or data mover that is already defined to the server.

#### *destination\_name* (Required)

Specifies the name of the destination of the path to be deleted. This parameter is required.

#### SRCType (Required)

Specifies the source type of the path to be deleted. This parameter is required. Possible values are:

##### DATAMover

Specifies that a data mover is the source.

##### SERVer

Specifies that a storage agent is the source.

#### DESTType (Required)

Specifies the type of the destination. Possible values are:

##### DRive LIBRARY=*library\_name*

Specifies that a drive is the destination. The DRIVE and LIBRARY parameters are both required when the destination type is drive.

##### LIBRARY

Specifies that a library is the destination.

**Attention:** If the path from a data mover to a library is deleted, or the path from the server to a library is deleted, the server will not be able to access the library. If the server is halted and restarted while in this state, the library will not be initialized.

### Example: Delete a NAS data mover path

Delete a path from a NAS data mover NAS1 to the library NASLIB.

```
delete path nas1 naslib srctype=datamover desttype=library
```

## DELETE PATH

### Related commands

*Table 132. Commands related to DELETE PATH*

| Command          | Description   |
|------------------|---|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server.      |
| DEFINE PATH      | Defines a path from a source to a destination.                      |
| QUERY PATH       | Displays information about the path from a source to a destination. |
| UPDATE PATH      | Changes the attributes associated with a path.                      |



## DELETE POLICYSET (Delete a policy set)

Use this command to delete a policy set. You cannot delete the ACTIVE policy set. When you delete a policy set, all management classes and copy groups that belong to the policy set are also deleted.

You can delete the predefined STANDARD policy set. However, if you later reinstall the Tivoli Storage Manager server, the process restores all STANDARD policy objects.

### 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—domain_name—policy_set_name—◄◄
```

### Parameters

*domain\_name* **(Required)**

Specifies the policy domain to which the policy set belongs.

*policy\_set\_name* **(Required)**

Specifies the policy set to delete.

### Example: Delete a policy set

Delete the VACATION policy set from the EMPLOYEE\_RECORDS policy domain.

#### Command

```
delete policyset employee_records vacation
```

### Related commands

Table 133. Commands related to DELETE POLICYSET

| Command            | Description  |
|--------------------|--|
| ACTIVATE POLICYSET | Validates and activates a policy set.  |
| COPY POLICYSET     | Creates a copy of a policy set.  |
| DEFINE POLICYSET   | Defines a policy set within the specified policy domain.   |
| QUERY POLICYSET    | Displays information about policy sets.  |
| UPDATE POLICYSET   | Changes the description of a policy set.   |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

## DELETE PROFASSOCIATION (Delete a profile association)

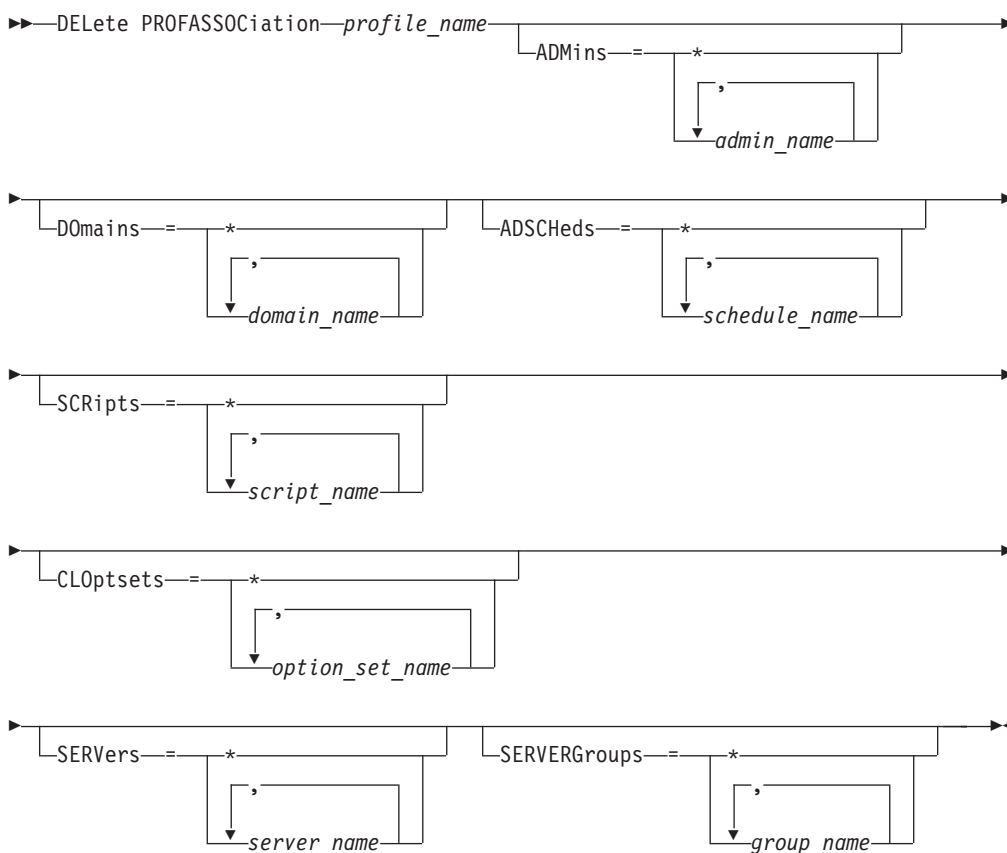
Use this command on a configuration manager to delete the association of one or more objects from a profile. If associations are deleted, the objects are no longer distributed to subscribing managed servers. When managed servers request updated configuration information, the configuration manager notifies them of the object deletions.

A managed server deletes the objects that were deleted from the profile, unless the objects are associated with another profile to which that server subscribes.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *profile\_name* (Required)

Specifies the profile from which to delete associations.

#### ADMins

Specifies the administrators whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all character (\*) to delete all administrators from the profile. If you specify a list of administrators and a match-all definition exists for the profile, the command fails.

Administrator definitions are not changed on the configuration manager. However, they are automatically deleted from all subscribing managed servers at the next configuration refresh, with the following exceptions:

- An administrator is not deleted if that administrator has an open session on the server.
- An administrator is not deleted if, as a result, the managed server would have no administrators with system privilege class.

**DOmains**

Specifies the domains whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all character (\*) to delete all domains from the profile. If you specify a list of domains and a match-all domain definition exists for the profile, the command fails.

The domain information is automatically deleted from all subscribing managed servers. However, a policy domain that has client nodes assigned will not be deleted. To delete the domain at the managed server, assign those client nodes to another policy domain.

**ADScheds**

Specifies a list of administrative schedules whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. If you specify a list of administrative schedules and a match-all administrative schedule definition exists for the profile, the command fails. Use the match-all character (\*) to delete all administrative schedules from the profile.

The administrative schedules are automatically deleted from all subscribing managed servers. However, an administrative schedule is not deleted if the schedule is active on the managed server. To delete an active schedule, make the schedule inactive.

**SCRipts**

Specifies the server command scripts whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all character (\*) to delete all scripts from the profile. If you specify a list of scripts and a match-all script definition exists for the profile, the command fails. The server command scripts are automatically deleted from all subscribing managed servers.

**CLOptsets**

Specifies the client option sets whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. Use the match-all character (\*) to delete all client option sets from the profile. If you specify a list of client option sets and a match-all client option set definition exists for the profile, the command fails. The client option sets are automatically deleted from all subscribing managed servers.

**SERVers**

Specifies the servers whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. You can use the match-all character (\*) to delete all servers from the profile. If you specify a list of servers and a match-all server definition exists for the profile, the command fails. The server definitions are automatically deleted from all subscribing managed servers with the following exceptions:

## DELETE PROFASSOCIATION

- A server definition is not deleted if the managed server has an open connection to another server.
- A server definition is not deleted if the managed server has a device class of the device type SERVER that refers to the other server.
- A server definition is not deleted if the server is the event server for the managed server.

### SERVERGroups

Specifies the server groups whose association with the profile is deleted. You can specify more than one name by separating the names with commas and no intervening spaces. You can use the match-all character (\*) to delete all server groups from the profile. If you specify a list of server groups and a match-all group definition exists for the profile, the command fails. The server group definitions are automatically deleted from all subscribing managed servers.

### Example: Delete the domain associations for a specific profile

Delete all domain associations from a profile named MIKE.

```
delete profassociation mike domains=*
```

### Related commands

Table 134. Commands related to DELETE PROFASSOCIATION

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                       |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                 |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                    |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                  |
| NOTIFY SUBSCRIBERS     | Notifies servers to refresh their configuration information.       |
| QUERY PROFILE          | Displays information about configuration profiles.                 |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |
| UNLOCK PROFILE         | Enables a locked profile to be distributed to managed servers.     |
| UPDATE PROFILE         | Changes the description of a profile.                              |

DELETE PROFILE (Delete a profile)

Use this command on a configuration manager to delete a profile and stop its distribution to managed servers.

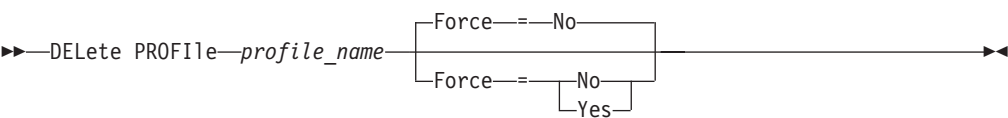
You cannot delete a locked profile. You must first unlock the profile with the UNLOCK PROFILE command.

Deleting a profile from a configuration manager does not delete objects associated with that profile from the managed servers. You can use the DELETE SUBSCRIPTION command with the DISCARDOBJECTS=YES parameter on each subscribing managed server to delete subscriptions to the profile and associated objects. This also prevents the managed servers from requesting further updates to the profile.

Privilege class

To issue this command, you must have system privilege.

Syntax



Parameters

profile\_name (Required)  
Specifies the profile to delete.

Force  
Specifies whether the profile is deleted if one or more managed servers have subscriptions to that profile. The default is NO. Possible values are:

No  
Specifies that the profile is not deleted if one or more managed servers have subscriptions to that profile. You can delete the subscriptions on each managed server using the DELETE SUBSCRIPTION command.

Yes  
Specifies that the profile is deleted even if one or more managed servers have subscriptions to that profile. Each subscribing server continues to request updates for the deleted profile until the subscription is deleted.

Examples: Delete a profile

Delete a profile named BETA, even if one or more managed servers subscribe to it.  
delete profile beta force=yes

Related commands

Table 135. Commands related to DELETE PROFILE

| Command                | Description                        |
|------------------------|------------------------------------|
| COPY PROFILE           | Creates a copy of a profile.       |
| DEFINE PROFASSOCIATION | Associates objects with a profile. |

## DELETE PROFILE

*Table 135. Commands related to DELETE PROFILE (continued)*

| Command                | Description  |
|------------------------|--|
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DEFINE SUBSCRIPTION    | Subscribes a managed server to a profile.                          |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| DELETE SUBSCRIPTION    | Deletes a specified profile subscription.                          |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                  |
| QUERY PROFILE          | Displays information about configuration profiles.                 |
| QUERY SUBSCRIPTION     | Displays information about profile subscriptions.                  |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |
| UNLOCK PROFILE         | Enables a locked profile to be distributed to managed servers.     |
| UPDATE PROFILE         | Changes the description of a profile.                              |

## DELETE RECMEDMACHASSOCIATION (Delete recovery media and machine association)

Use this command to remove the association of one or more machines with a recovery media. This command does not delete the machine from Tivoli Storage Manager.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶—DELEte RECMEDMACHAssociation—media_name—machine_name—▶▶

```

### Parameters

#### *media\_name* (Required)

Specifies the name of the recovery media that is associated with one or more machines.

#### *machine\_name* (Required)

Specifies the name of the machine associated with the recovery media. To specify a list of machine names, separate the names with commas and no intervening spaces. You can use wildcard characters to specify a name. If a machine is not associated with the recovery media, the machine is ignored.

### Example: Delete a machine's association with recovery media

Delete the association between the DIST5RM recovery media and the DISTRICT1 and DISTRICT5 machines.

```

delete recmedmachassociation
dist5rm district1,district5

```

### Related commands

Table 136. Commands related to DELETE RECMEDMACHASSOCIATION

| Command                      | Description                                    |
|------------------------------|--|
| DEFINE RECMEDMACHASSOCIATION | Associates recovery media with a machine.      |
| QUERY MACHINE                | Displays information about machines.           |
| QUERY RECOVERYMEDIA          | Displays media available for machine recovery. |

## DELETE RECOVERYMEDIA (Delete recovery media)

Use this command to delete a recovery media definition from Tivoli Storage Manager.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—DELEte RECOVERYMedia—*media\_name*—◄◄

### Parameters

*media\_name* **(Required)**

Specifies the name of the recovery media.

### Example: Delete a recovery media definition

Delete the DIST5RM recovery media.

```
delete recoverymedia dist5rm
```

### Related commands

Table 137. Commands related to DELETE RECOVERYMEDIA

| Command              | Description                                      |
|----------------------|--|
| DEFINE RECOVERYMEDIA | Defines the media required to recover a machine. |
| QUERY RECOVERYMEDIA  | Displays media available for machine recovery.   |
| UPDATE RECOVERYMEDIA | Changes the attributes of recovery media.        |



## DELETE SCHEDULE (Delete a client or an administrative command schedule)

Use this command to delete schedules from the database.

The DELETE SCHEDULE command takes two forms: one if the schedule applies to client operations, one if the schedule applies to administrative commands. The syntax and parameters for each form are defined separately.

*Table 138. Commands related to DELETE SCHEDULE*

| Command         | Description   |
|-----------------|---|
| COPY SCHEDULE   | Creates a copy of a schedule.   |
| DEFINE SCHEDULE | Defines a schedule for a client operation or an administrative command. |
| QUERY SCHEDULE  | Displays information about schedules.                                   |
| UPDATE SCHEDULE | Changes the attributes of a schedule.                                   |

## DELETE SCHEDULE

### DELETE SCHEDULE (Delete a client schedule)

Use the DELETE SCHEDULE command to delete one or more client schedules from the database. Any client associations to a schedule are removed when the schedule is deleted.

#### Privilege class

To delete a client schedule, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the specified policy domain.

#### Syntax

```
►►—DELeTe SChedule—domain_name—schedule_name—Type=Client—►◄
```

#### Parameters

*domain\_name* **(Required)**

Specifies the name of the policy domain to which the schedule belongs.

*schedule\_name* **(Required)**

Specifies the name of the schedule to delete. You can use a wildcard character to specify this name.

**Type=Client**

Specifies to delete a client schedule. This parameter is optional. The default is CLIENT.

#### Example: Delete a specific schedule from a specific policy domain

Delete the WEEKLY\_BACKUP schedule, which belongs to the EMPLOYEE\_RECORDS policy domain.

```
delete schedule employee_records weekly_backup
```

**DELETE SCHEDULE (Delete an administrative schedule)**

Use this command to delete one or more administrative command schedules from the database.

**Privilege class**

To delete an administrative command schedule, you must have system authority.

**Syntax**

```
►►—DELEte SChedule—schedule_name—Type—=—Administrative—————►◄
```

**Parameters*****schedule\_name* (Required)**

Specifies the name of the schedule to delete. You can use a wildcard character to specify this name.

**Type=Administrative (Required)**

Specifies to delete an administrative command schedule.

**Example: Delete an administrative command schedule**

Delete the administrative command scheduled named DATA\_ENG.

```
delete schedule data_eng type=administrative
```

## DELETE SCRIPT (Delete command lines from a script or delete the entire script)

Use this command to delete a single line from a Tivoli Storage Manager script or to delete the entire Tivoli Storage Manager script.

### Privilege class

To issue this command, the administrator must have previously defined the script or must have system privilege.

### Syntax

```
►► DELEte SCript—script_name—┐
                               └─Line—=number—┘
```

### Parameters

#### *script\_name* (Required)

Specifies the name of the script to delete. The script is deleted unless you specify a line number.

#### Line

Specifies the line number to delete from the script. If you do not specify a line number, the entire script is deleted.

### Example: Delete a specific line from a script

Using the following script named QSAMPLE and issue a command to delete line 005 from it.

```
001  /* This is a sample script */
005  QUERY STATUS
010  QUERY PROCESS

delete script qsampl line=5
```

### Related commands

Table 139. Commands related to DELETE SCRIPT

| Command       | Description  |
|---------------|--|
| COPY SCRIPT   | Creates a copy of a script.                                |
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| QUERY SCRIPT  | Displays information about scripts.                        |
| RENAME SCRIPT | Renames a script to a new name.                            |
| RUN           | Runs a script.   |
| UPDATE SCRIPT | Changes or adds lines to a script.                         |

## DELETE SERVER (Delete a server definition)

Use this command to delete a server definition.

This command fails if the server:

- Is defined as the event server.
- Is named in a device class definition whose device type is SERVER.
- Has an open connection to or from another server.
- Is a target server for virtual volumes.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—DELEte—SERver—*server\_name*—◄◄

### Parameters

*server\_name* (Required)

Specifies a server name.

### Example: Delete a server's definition

Delete the definition for a server named SERVER2.

```
delete server server2
```

### Related commands

Table 140. Commands related to DELETE SERVER

| Command           | Description  |
|-------------------|--|
| DEFINE SERVER     | Defines a server for server-to-server communications.                                  |
| QUERY EVENTSERVER | Displays the name of the event server.   |
| QUERY SERVER      | Displays information about servers.  |
| RECONCILE VOLUMES | Reconciles source server virtual volume definitions and target server archive objects. |
| UPDATE SERVER     | Updates information about a server.  |

## DELETE SERVERGROUP (Delete a server group)

Use this command to delete a server group. If the group you delete is a member of other server groups, Tivoli Storage Manager also removes the group from the other groups.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—DELEte SERVERGRoup—*group\_name*—◄◄

### Parameters

*group\_name* **(Required)**

Specifies the server group to delete.

### Example: Delete a server group

Delete a server group named WEST\_COMPLEX.

```
delete servergroup west_complex
```

### Related commands

Table 141. Commands related to DELETE SERVERGROUP

| Command            | Description                                     |
|--------------------|---|
| COPY SERVERGROUP   | Creates a copy of a server group.               |
| DEFINE GRPMEMBER   | Defines a server as a member of a server group. |
| DEFINE SERVERGROUP | Defines a new server group.                     |
| DELETE GRPMEMBER   | Deletes a server from a server group.           |
| MOVE GRPMEMBER     | Moves a server group member.                    |
| QUERY SERVERGROUP  | Displays information about server groups.       |
| RENAME SERVERGROUP | Renames a server group.                         |
| UPDATE SERVERGROUP | Updates a server group.                         |

## DELETE SPACETRIGGER (Delete the storage pool space triggers)

Use this command to delete the definition of the storage pool space trigger.

### Privilege class

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

### Syntax

```
►►—DELEte SPACETriGger—STG—┐
                             └STGPOOL—==—storage_pool_name—┘
```

### Parameters

#### STG

Specifies a storage pool space trigger

#### STGPOOL

Specifies the storage pool trigger to be deleted. If STG is specified without specifying STGPOOL, the default storage pool space trigger is the deletion target.

### Example: Delete a space trigger definition

Delete the space trigger definition for the WINPOOL1 storage pool.

```
delete spacetrigger stg stgpool=winpool1
```

### Related commands

Table 142. Commands related to DELETE SPACETRIGGER

| Command             | Description   |
|---------------------|---|
| DEFINE SPACETRIGGER | Defines a space trigger to expand the space for a storage pool. |
| QUERY SPACETRIGGER  | Displays information about a storage pool space trigger.        |
| UPDATE SPACETRIGGER | Changes attributes of storage pool space trigger.               |

## DELETE STGPOOL (Delete a storage pool)

Use this command to delete a storage pool. To delete a storage pool, you must first delete all volumes assigned to the storage pool.

You cannot delete a storage pool that is identified as the next storage pool for another storage pool. For more information on storage pool hierarchy, see the NEXTSTGPOOL parameter in the DEFINE STGPOOL command.

### Important:

- Do not delete a storage pool that is specified as a destination for a management class or copy group in the ACTIVE policy set. Client operations might fail as a result.
- When deleting a copy storage pool that has been previously included in a primary storage-pool definition (specifically in the COPYSTGPOOLS list), you must remove the copy storage pool from the list prior to deletion. Otherwise, the DELETE STGPOOL command fails until all references to that copy pool have been removed. For each primary storage pool with a reference to the copy storage pool to be deleted, remove the reference by entering the UPDATE STGPOOL command with the COPYSTGPOOLS parameter with all previous copy storage pools except the copy storage pool to be deleted.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—DELEte STGpool—*pool\_name*—◄◄

### Parameters

#### *pool\_name* (Required)

Specifies the storage pool to delete.

### Example: Delete a storage pool

Delete the storage pool named POOLA.

```
delete stgpool poola
```

### Related commands

Table 143. Commands related to DELETE STGPOOL

| Command            | Description   |
|--------------------|---|
| BACKUP STGPOOL     | Backs up a primary storage pool to a copy storage pool.               |
| DEFINE STGPOOL     | Defines a storage pool as a named collection of server storage media. |
| QUERY STGPOOL      | Displays information about storage pools.                             |
| SET DRMCOPYSTGPOOL | Specifies that copy storage pools are managed by DRM.                 |
| UPDATE STGPOOL     | Changes the attributes of a storage pool.                             |



## DELETE SUBSCRIBER (Delete subscriptions from a configuration manager database)

Use this command on a configuration manager to delete managed server subscriptions from the configuration manager database. Use this command when a managed server no longer exists or cannot notify the configuration manager after deleting a subscription.

**Attention:** Use this command only in rare situations in which the configuration manager's database contains an entry for a subscription, but the managed server does not have such a subscription. For example, use this command if a managed server no longer exists or cannot notify the configuration manager after deleting a subscription.

Under normal circumstances, use the DELETE SUBSCRIPTION command to delete a subscription from the managed server. The managed server notifies the configuration manager, which then deletes the subscription from its database.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—DELEte SUBSCRIBer—server_name—————◄◄
```

### Parameters

#### *server\_name* (Required)

Specifies the name of the managed server with subscription entries to be deleted.

### Example: Delete subscription entries for a specific managed server

Delete all subscription entries for a managed server named DAN.

```
delete subscriber dan
```

### Related commands

Table 144. Commands related to DELETE SUBSCRIBER

| Command             | Description   |
|---------------------|---|
| DEFINE SUBSCRIPTION | Subscribes a managed server to a profile.                                   |
| DELETE SUBSCRIPTION | Deletes a specified profile subscription.                                   |
| NOTIFY SUBSCRIBERS  | Notifies servers to refresh their configuration information.                |
| QUERY SUBSCRIBER    | Displays information about subscribers and their subscriptions to profiles. |
| QUERY SUBSCRIPTION  | Displays information about profile subscriptions.                           |

## DELETE SUBSCRIPTION

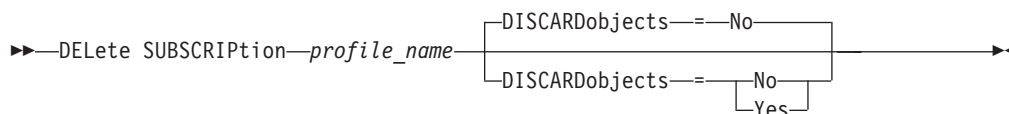
### DELETE SUBSCRIPTION (Delete a profile subscription)

Use this command on a managed server to delete a profile subscription. You can also delete from the managed server all objects associated with the profile.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax



#### Parameters

##### *profile\_name* (Required)

Specifies the name of the profile for which the subscription is to be deleted.

##### DISCARDobjects

Specifies whether objects associated with the profile are to be deleted on the managed server. This parameter is optional. The default is NO.

##### No

Specifies that the objects are not to be deleted.

##### Yes

Specifies that the objects are to be deleted, unless they are associated with another profile for which a subscription is defined.

#### Example: Delete a profile subscription

Delete a subscription to a profile named ALPHA and its associated objects from a managed server.

```
delete subscription alpha discardobjects=yes
```

#### Related commands

Table 145. Commands related to DELETE SUBSCRIPTION

| Command             | Description   |
|---------------------|---|
| DEFINE SUBSCRIPTION | Subscribes a managed server to a profile.                                   |
| DELETE SUBSCRIBER   | Deletes obsolete managed server subscriptions.                              |
| NOTIFY SUBSCRIBERS  | Notifies servers to refresh their configuration information.                |
| QUERY SUBSCRIBER    | Displays information about subscribers and their subscriptions to profiles. |
| QUERY SUBSCRIPTION  | Displays information about profile subscriptions.                           |

## DELETE VIRTUALFSMAPPING (Delete a virtual file space mapping)

Use this command to delete a virtual file space mapping definition. Virtual file spaces containing data cannot be deleted unless you use the DELETE FILESPACE command first.

### Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege
- Restricted policy privilege for the domain to which the NAS node is assigned

### Syntax

►►—DELEte VIRTUALFSmapping—*node\_name*—*virtual\_filespace\_name*—————►►

### Parameters

#### *node\_name* (Required)

Specifies the NAS node on which the file system and path reside. You cannot use wildcard characters or specify a list of names.

#### *virtual\_filespace\_name* (Required)

Specifies the name of the virtual file space mapping definition to be deleted. Wildcard characters are allowed.

### Example: Delete a virtual file space mapping

Delete the virtual file space mapping definition /mikesomedir for the NAS node named NAS1.

```
delete virtualfsmapping nas1 /mikesomedir
```

### Related commands

Table 146. Commands related to DELETE VIRTUALFSMAPPING

| Command                 | Description                          |
|-------------------------|--------------------------------------|
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping. |
| QUERY VIRTUALFSMAPPING  | Query a virtual file space mapping.  |
| UPDATE VIRTUALFSMAPPING | Update a virtual file space mapping. |

## DELETE VOLHISTORY (Delete sequential volume history information)

Use this command to delete volume history file records that are no longer needed (for example, records for obsolete database backup volumes).

When you delete records for volumes that are not in storage pools (for example, database backup or export volumes), the volumes return to scratch status even if Tivoli Storage Manager acquired them as private volumes. Scratch volumes of device type FILE are deleted. When you delete the records for storage pool volumes, the volumes remain in the Tivoli Storage Manager database. When you delete records for recovery plan file objects from a source server, the objects on the target server are marked for deletion.

Use the DELETE BACKUPSET command to delete specified backup set volume information in the volume history file. Do not use this DELETE VOLHISTORY command to delete backup set volume information in the volume history file.

For users of DRM, the database backup expiration should be controlled with the SET DRMDBBACKUPEXPIREDAYS command instead of this DELETE VOLHISTORY command. Using the DELETE VOLHISTORY command removes Tivoli Storage Manager's record of the volume. This can cause volumes to be lost that were managed by the MOVE DRMEDIA command. The recommended way to manage the automatic expiration of DRM database backup volumes is by using the SET DRMDBBACKUPEXPIREDAYS command.

### Notes:

1. Volumes for the most recent database backup series are not deleted.
2. Existing volume history files are not automatically updated with this command.
3. You can use the DEFINE SCHEDULE command to periodically delete volume history records.

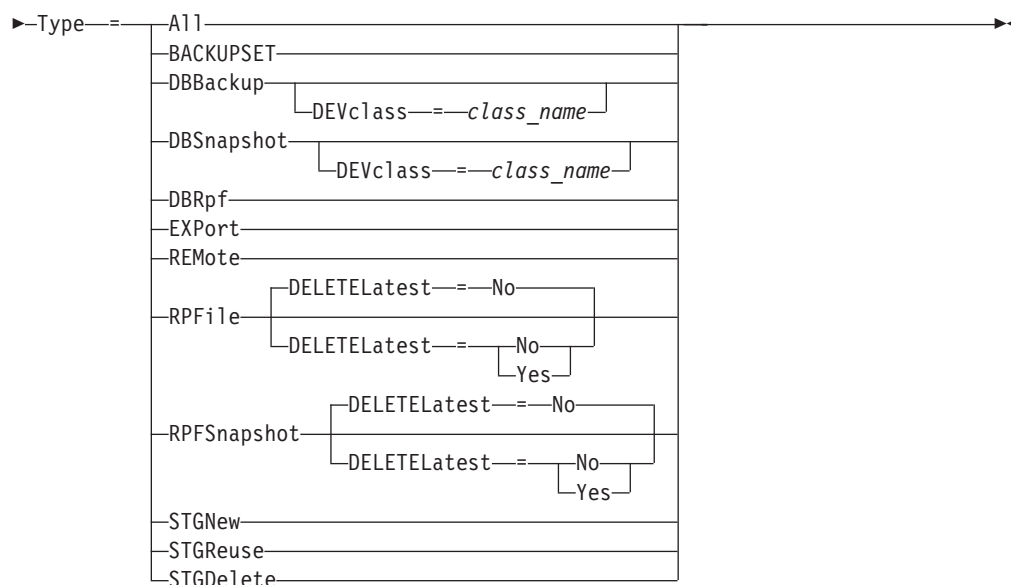
### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶ DELeTe VOLHistory—TODate—date — TTime—23:59:59  
TTime—time →
  
```



## Parameters

### TODate (Required)

Specifies the date to use to select sequential volume history information to be deleted. Tivoli Storage Manager deletes only those records with a date on or before the date you specify. You can specify the date using one of the values below:

| Value               | Description  | Example   |
|---------------------|--|---|
| MM/DD/YYYY          | A specific date  | 01/23/1999  |
| TODAY               | The current date   | TODAY   |
| TODAY-days or -days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY-30 or -30.<br><br>To delete records that are 30 or more days old, you can specify TODAY-30 or simply -30. |

### TOTime

Specifies that you want to delete records created on or before this time on the specified date. This parameter is optional. The default is the end of the day (23:59:59). You can specify the time using one of the values below:

| Value    | Description                            | Example  |
|----------|--|----------|
| HH:MM:SS | A specific time on the specified date  | 12:30:22 |
| NOW      | The current time on the specified date | NOW      |

## DELETE VOLHISTORY

| Value                  | Description  | Example  |
|------------------------|--|--|
| NOW+HH:MM or<br>+HH:MM | The current time plus hours and minutes on the specified date  | NOW+03:00 or +03:00.<br><br>If you issue the DELETE VOLHISTORY command at 9:00 with TOTIME=NOW+03:00 or TOTIME=+03:00, Tivoli Storage Manager deletes records with a time of 12:00 or earlier on the specified date. |
| NOW-HH:MM or<br>-HH:MM | The current time minus hours and minutes on the specified date | NOW-03:30 or -03:30.<br><br>If you issue the DELETE VOLHISTORY command at 9:00 with TOTIME=NOW-3:30 or TOTIME=-3:30, Tivoli Storage Manager deletes records with a time of 5:30 or earlier on the specified date.    |

### Type (Required)

Specifies the type of records, which also meet the date and time criteria, to delete from the volume history file. Possible values are:

#### All

Specifies to delete all records.

#### BACKUPSET

Specifies to delete all backup set volumes.

#### DBBackup

Specifies to delete only records that contain information about volumes used for database full and incremental backups, that is with volume types of BACKUPFULL and BACKUPINCR, and that meet the specified date and time criteria. The latest database full and incremental backup series will not be deleted.

#### DEVclass=*class\_name*

Specifies the device class name that was used to create the database backups. This optional parameter can be used to delete database backups created using a server-to-server virtual volume device class. The type of the device class must be SERVER. This parameter can only be used to delete volume history entries of type BACKUPFULL, BACKUPINCR, or DBSNAPSHOT.

A full, incremental, or snapshot database backup volume is eligible to be deleted if all of the following conditions are met:

- The device class used to create the database backup volume matches the specified device class
- The volume was created on or before the specified date and time
- The volume is not part of the latest full plus incremental database backup series if the specified volume type is DBBackup, or snapshot database backup series if the volume type is DBSnapshot

#### DBSnapshot

Specifies to delete only records that contain information about volumes used for snapshot database backups, and that meet the specified date and time criteria. The latest snapshot database backup will not be deleted.

**DEVclass=classname**

Specifies the device class name that was used to create the database backups. This optional parameter can be used to delete database backups created using a server-to-server virtual volume device class. The type of the device class must be SERVER. This parameter can only be used to delete volume history entries of type BACKUPFULL, BACKUPINCR, or DBSNAPSHOT.

A full, incremental, or snapshot database backup volume is eligible to be deleted if all of the following conditions are met:

- The device class used to create the database backup volume matches the specified device class
- The volume was created on or before the specified date and time
- The volume is not part of the latest full plus incremental database backup series if the specified volume type is DBBackup, or snapshot database backup series if the volume type is DBSnapshot

**DBRpf**

Specifies to delete only records that contain information about full and incremental database backup volumes and recovery plan file volumes.

**EXPort**

Specifies to delete only records that contain information about export volumes.

**Remote**

Specifies to delete only records that contain information about volumes used by library clients.

**RPFfile**

Specifies to delete only records that contain information about recovery plan file objects that are stored on a target server and that meet the specified date and time criteria.

**DELETEDatest**

Specifies whether the latest recovery plan file is eligible for deletion. This optional parameter can be used to delete the latest recovery plan files created using a server-to-server virtual volume device class.

This parameter can only be used to delete volume history entries of type RPFfile (for instance, those recovery plan files that were created using the DEVCLASS parameter with the PREPARE command). If this parameter is not specified, the latest RPFfile entries are not deleted.

**No** Specifies the latest RPFfile file is not deleted.

**Yes** Specifies the latest RPFfile file is deleted if it meets the specified date and time criteria.

**RPFSnapshot**

Specifies to delete only records that contain information about recovery plan file objects that were created assuming snapshot database backups, that are stored on a target server and that meet the specified date and time criteria. The latest RPFsnapshot file will not be deleted unless it meets the specified date and time criteria, and the DELETEDatest parameter is set to Yes.

**DELETEDatest**

Specifies whether the latest recovery plan file is eligible for deletion. This optional parameter can be used to delete the latest recovery plan files created using a server-to-server virtual volume device class.

## DELETE VOLHISTORY

This parameter can only be used to delete volume history entries of type RPFSNAPSHOT (for instance, those recovery plan files that were created using the DEVCLASS parameter with the PREPARE command). If this parameter is not specified, the latest RPFSNAPSHOT entries are not deleted.

**No** Specifies the latest RPFSNAPSHOT file is not deleted.

**Yes** Specifies the latest RPFSNAPSHOT file is deleted if it meets the specified date and time criteria.

### STGNew

Specifies to delete only records that contain information about new sequential access storage volumes.

### STGReuse

Specifies to delete only records that contain information about reused sequential storage pool volumes.

### STGDelete

Specifies to delete only records that contain information about deleted sequential storage pool volumes.

## Example: Delete recovery plan file information

Delete all recovery plan file information created on or before 03/28/2005.

```
delete volhistory type=rpfile todate=03/28/2005
```

## Related commands

Table 147. Commands related to DELETE VOLHISTORY

| Command                   | Description   |
|---------------------------|---|
| BACKUP VOLHISTORY         | Records volume history information in external files.                                 |
| DEFINE SCHEDULE           | Defines a schedule for a client operation or an administrative command.               |
| DELETE VOLUME             | Deletes a volume from a storage pool.   |
| EXPIRE INVENTORY          | Manually starts inventory expiration processing.                                      |
| MOVE DRMEDIA              | Moves DRM media on-site and off-site.   |
| PREPARE                   | Creates a recovery plan file.   |
| QUERY RPFILE              | Displays information about recovery plan files.                                       |
| QUERY VOLHISTORY          | Displays sequential volume history information that has been collected by the server. |
| SET DRMRPFEXPIREDAYS      | Set criteria for recovery plan file expiration.                                       |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.                             |



## DELETE VOLUME (Delete a storage pool volume)

Use this command to delete a storage pool volume and, optionally, the files stored in the volume.

If the volume has data, to delete the volume you must do one of the following:

- Before deleting the volume, use the MOVE DATA command to move all files to another volume.
- Explicitly request to discard all files in the volume when the volume is deleted (by specifying DISCARDDATA=YES).

If you are deleting several volumes, delete the volumes one at a time. Deleting more than one volume at a time can adversely affect server performance.

Storage pool volumes cannot be deleted if they are in use. For example, a volume cannot be deleted if a user is restoring or retrieving a file residing in the volume, if the server is writing information to the volume, or if a reclamation process is using the volume.

If this command is applied to a WORM (write once, read many) volume, the volume returns to scratch if it has space remaining in which data can be written. (Note that data on WORM volumes, including deleted and expired data, cannot be overwritten. Therefore, data can only be written in space that does not contain current, deleted, or expired data.) If a WORM volume does not have any space available in which data can be written, it remains private. To remove the volume from the library, you must use the CHECKOUT LIBVOLUME command.

The DELETE VOLUME command automatically updates the server library inventory for sequential volumes if the volume is returned to scratch status when the volume becomes empty. To determine whether a volume will be returned to scratch status, issue the QUERY VOLUME command and look at the output. If the value for the attribute "Scratch Volume?" is "Yes," then the server library inventory is automatically updated.

If the value is "No," you can issue the UPDATE LIBVOLUME command to specify the status as scratch. It is recommended that you issue the UPDATE LIBVOLUME command after issuing the DELETE VOLUME command.

Attempting to use the DELETE VOLUME command to delete WORM FILE volumes in a storage pool with RECLAMATIONTYPE=SNAPLOCK fails with an error message. Deletion of empty WORM FILE volumes is performed only by the reclamation process.

If you issue the DELETE VOLUME command for a volume in a storage pool that has a SHRED parameter value greater than 0, the volume is placed in the pending state until shredding is run. Shredding is necessary to complete the deletion, even if the volume is empty.

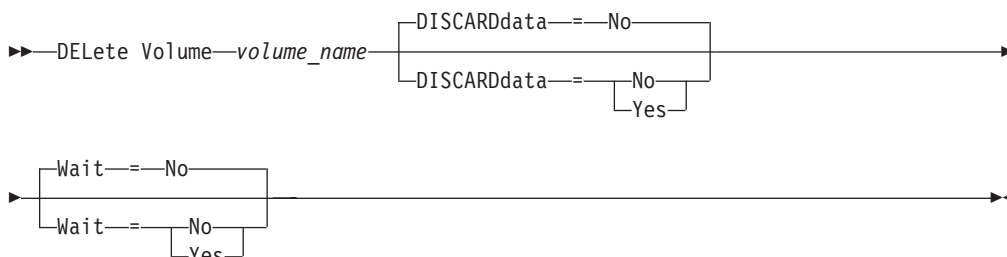
If you issue the DELETE VOLUME command for a volume in a storage pool that is set up for deduplication, the Tivoli Storage Manager destroys any object that is referencing data on that volume.

## DELETE VOLUME

### 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.

### Syntax



### Parameters

#### *volume\_name* (Required)

Specifies the name of the volume to delete.

#### DISCARDdata

Specifies whether files stored in the volume are deleted. This parameter is optional. The default value is NO. Possible values are:

##### No

Specifies that files stored in the volume are not deleted. If the volume contains any files, the volume is not deleted.

##### Yes

Specifies that all files stored in the volume are deleted. The server does not need to mount the volume for this type of deletion.

#### Note:

1. The Tivoli Storage Manager server does not delete archive files that are on deletion hold.
2. If archive retention protection is enabled, the Tivoli Storage Manager server deletes only archive files whose retention period has expired.

If the volume being deleted is a primary storage pool volume, the server checks whether any copy storage pool has copies of files that are being deleted. When files stored in a primary storage pool volume are deleted, any copies of these files in copy storage pools are also deleted.

When you delete a disk volume in a primary storage pool, the command also deletes any files that are cached copies (copies of files that have been migrated to the next storage pool). Deleting cached copies of files does not delete the files that have already been migrated or backed up to copy storage pools. Only the cached copies of the files are affected.

If the volume being deleted is a copy storage pool volume, only files on the copy pool volume are deleted. The primary storage pool files are not affected.

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

If you cancel the DELETE VOLUME operation during processing or if a system failure occurs, some files might remain on the volume. You can delete the same volume again to have the server delete the remaining files and then the volume.

#### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter affects processing only when you have also requested that any data on the volume be discarded. This parameter is optional. The default value is No. Possible values are:

#### No

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

The server displays messages that are created from the background process either in the activity log or the server console, depending on where messages are logged.

#### Yes

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Note:** You cannot specify WAIT=YES from the server console.

### Example: Delete a storage pool volume

Delete storage pool volume stgvol.1 from the storage pool FILEPOOL.

```
delete volume stgvol.1
```

### Related commands

Table 148. Commands related to DELETE VOLUME

| Command        | Description   |
|----------------|---|
| CANCEL PROCESS | Cancels a background server process.  |
| DEFINE VOLUME  | Assigns a volume to be used for storage within a specified storage pool.        |
| MOVE DATA      | Moves data from a specified storage pool volume to another storage pool volume. |
| MOVE DRMEDIA   | Moves DRM media on-site and off-site.   |
| QUERY CONTENT  | Displays information about files in a storage pool volume.                      |
| QUERY DRMEDIA  | Displays information about disaster recovery volumes.                           |
| QUERY PROCESS  | Displays information about background processes.                                |
| QUERY VOLUME   | Displays information about storage pool volumes.                                |
| UPDATE VOLUME  | Updates the attributes of storage pool volumes.                                 |

---

### DISABLE commands

Use DISABLE commands to prevent some types of operations by the server.

The following is a list of DISABLE commands for Tivoli Storage Manager:

- “DISABLE EVENTS (Disable events for event logging)” on page 407
- “DISABLE SESSIONS (Temporarily prevent client node access to the server)” on page 411

DISABLE EVENTS (Disable events for event logging)

Use this command to disable the processing of one or more events. If you specify a receiver that is not supported on any platform, or if you specify an invalid event or name, Tivoli Storage Manager issues an error message. However, any valid receivers, events, or names that you specified are still enabled.

**Tip:** Messages in the SEVERE category and message ANR9999D can provide valuable diagnostic information if there are serious server problems. For this reason, you should not disable these messages.

Restriction:

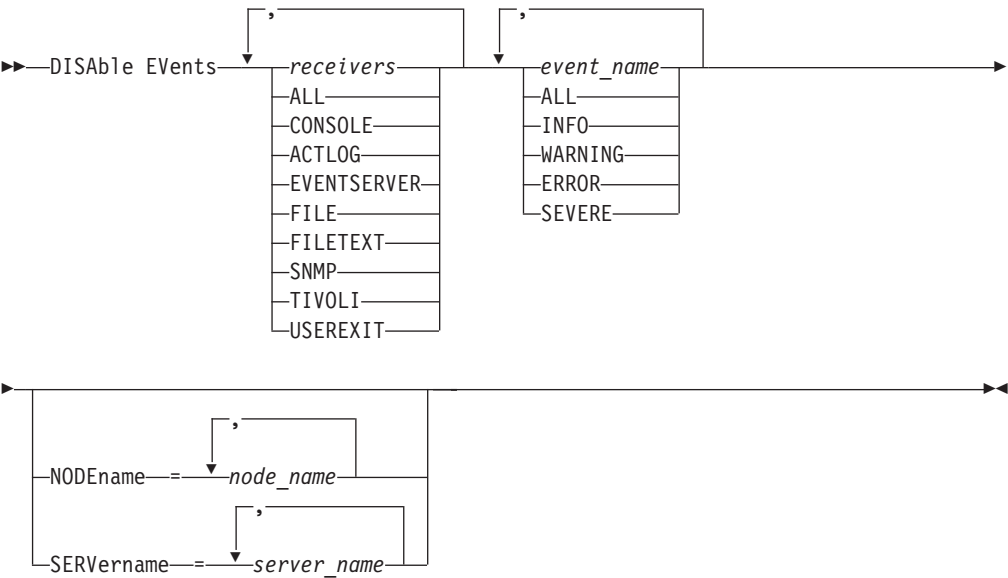
- Certain messages are displayed on the console even if they are disabled. These include some messages issued during server startup and shutdown and responses to administrative commands.
- Server messages from the server on which this command is issued cannot be disabled for the activity log.

ANR1822I indicates that event logging is being ended for the specified receiver. When the DISABLE EVENTS command is issued, this message is logged to the receiver even if it is one of the events that has been disabled. This is done to confirm that event logging has ended to that receiver, but subsequent ANR1822I messages are not logged to that receiver.

Privilege class

To issue this command, you must have system privilege.

Syntax



### Parameters

#### *receivers* (Required)

Specifies the name of the receivers for which to disable events. Specify multiple receivers by separating them with commas and no intervening spaces. Possible values are:

##### **ALL**

All receivers, except for server events on the activity log receiver (ACTLOG). Only client events can be disabled for the activity log receiver.

##### **CONSOLE**

The standard server console as a receiver.

##### **ACTLOG**

The activity log as a receiver. You can disable only client events, not server events, for the activity log.

##### **EVENTSERVER**

The event server as a receiver.

##### **FILE**

A user file as a receiver. Each logged event is a record in the file. The records are not easily readable by people.

##### **FILETEXT**

A user file as a receiver. Each logged event is a fixed-size, readable line.

##### **NTEVENTLOG**

The Windows application log as a receiver.

##### **SNMP**

The simple network management protocol (SNMP) as a receiver.

##### **TIVOLI**

The Tivoli Enterprise Console® (TEC) as a receiver.

##### **USEREXIT**

A user-written program as a receiver. The server writes information to the program.

#### *events* (Required)

Specifies the events to be disabled. You can specify multiple events by separating them with commas and no intervening spaces. Possible values are:

##### **ALL**

All events.

##### *event\_name*

A four-digit message number preceded by ANR for a server event or ANE for a client event. Valid ranges are from ANR0001 to ANR9999 and from ANE4000 to ANE4999. Specify the NODENAMES parameter if client events are to be disabled for matching nodes. Specify the SERVERNAME parameter if server events are to be disabled for matching servers.

For the TIVOLI event receiver only, you can specify the following events names for the IBM Tivoli Storage Manager application clients:

| Tivoli Storage Manager application client     | Prefix | Range     |
|---|--------|-----------|
| Data Protection for Microsoft Exchange Server | ACN    | 3500–3649 |
| Data Protection for Lotus Domino®             | ACD    | 5200–5299 |

| Tivoli Storage Manager application client | Prefix | Range     |
|---|--------|-----------|
| Data Protection for Oracle                | ANS    | 500–599   |
| Data Protection for Informix <sup>®</sup> | ANS    | 600–699   |
| Data Protection for Microsoft SQL Server  | ACO    | 3000–3999 |

**Remember:** Specifying ALL disables these messages. However, the INFO, WARNING, ERROR, and SEVERE options have no effect on the messages.

#### *severity categories*

If the event list contains a severity category, all events of that severity are disabled for the specified nodes. The message types are:

#### **INFO**

Information messages (type of I).

#### **WARNING**

Warning messages (type of W).

#### **ERROR**

Error messages (type of E).

#### **SEVERE**

Severe error messages (type of S).

#### **NODENAME**

Specifies the name of one or more node names for which events are to be disabled. You can use the wildcard character (\*) to specify all nodes. You can specify NODENAME or SERVERNAME. If neither parameter is specified, the events are disabled for the server running this command.

#### **SERVername**

Specifies the name of one or more server names for which events are to be disabled. You can use the wildcard character (\*) to specify all servers other than the server running this command. You can specify NODENAME or SERVERNAME. If neither parameter is specified, the events are disabled for the server running this command.

### **Example: Disable specific categories of events**

Disable all client events in the INFO and WARNING categories for the activity log and console receivers for all nodes.

```
disable events actlog,console
info,warning nodename=*
```

### **Related commands**

*Table 149. Commands related to DISABLE EVENTS*

| Command            | Description  |
|--------------------|--|
| BEGIN EVENTLOGGING | Starts event logging to a specified receiver.                  |
| ENABLE EVENTS      | Enables specific events for receivers.                         |
| END EVENTLOGGING   | Ends event logging to a specified receiver.                    |
| QUERY ENABLED      | Displays enabled or disabled events for a specific receiver.   |
| QUERY EVENTRULES   | Displays information about rules for server and client events. |

## DISABLE EVENTS

*Table 149. Commands related to DISABLE EVENTS (continued)*

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |



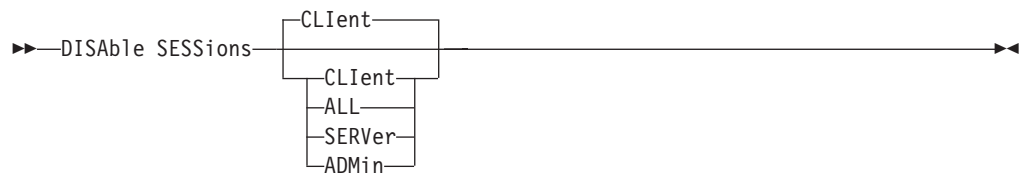
## DISABLE SESSIONS (Temporarily prevent client node access to the server)

Use this command to prevent any new sessions; however, active sessions will complete. Server processes, such as migration and reclamation, are not affected when you issue this command.

### Privilege class

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

### Syntax



### Parameters

Specifies the type of session to be disabled. This parameter is optional. The default value is CLIENT. Possible values are:

#### CLIENT

Disables only backup and archive client sessions.

#### ALL

Disables all session types.

#### SERVer

Disables only server-to-server sessions. These are the only types of sessions that are disabled:

- Server to server event logging
- Enterprise management
- Server registration
- LAN-free: storage agent - server
- Virtual volumes

#### ADMin

Disables only administrative sessions.

### Example: Prevent new client node backup and archive sessions on the server

Temporarily prevent new client node sessions from accessing the server.

```
disable sessions
```

### Example: Prevent all new sessions on the server

Temporarily prevent any new sessions from accessing the server.

```
disable sessions all
```

## DISABLE SESSIONS

### Related commands

*Table 150. Commands related to DISABLE SESSIONS*

| Command         | Description  |
|-----------------|--|
| CANCEL SESSION  | Cancels active sessions with the server.   |
| ENABLE SESSIONS | Resumes server activity following the DISABLE command or the ACCEPT DATE command.                        |
| QUERY SESSION   | Displays information about all active administrator and client sessions with IBM Tivoli Storage Manager. |
| QUERY STATUS    | Displays the settings of server parameters, such as those selected by the SET commands.                  |

---

## DISMOUNT command

Use the DISMOUNT command to dismount a volume by the real device address or by volume name.

The following is the DISMOUNT command for Tivoli Storage Manager:

- “DISMOUNT VOLUME (Dismount a volume by volume name)” on page 414

### DISMOUNT VOLUME (Dismount a volume by volume name)

Use this command to dismount an idle volume by volume name. If a drive cannot dismount the volume, manual intervention is required.

#### Privilege class

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

#### Syntax

►►—DISMount Volume—*volume\_name*—◄◄

#### Parameters

*volume\_name* (**Required**)

Specifies the name of the volume to dismount.

#### Example: Dismount a specific volume

Dismount the volume BTV005.

```
dismount volume btv005
```

#### Related commands

Table 151. Command related to DISMOUNT VOLUME

| Command     | Description   |
|-------------|---|
| QUERY MOUNT | Displays information about mounted sequential access media. |

## DISPLAY OBJNAME (Display a full object name)

Use this command when you want Tivoli Storage Manager to display a full object name if the name displayed in a message or query output has been abbreviated due to length. Object names that are very long can be difficult to display and use through normal operating system facilities. The Tivoli Storage Manager server will abbreviate long names and assign them a token ID which might be used if the object path name exceeds 1024 bytes. The token ID is displayed in a string that includes identifiers for the node, filesystem, and object name. The format is: [TSMOBJ:*nID.fsID.objID*]. When specified with the DISPLAY OBJNAME command, the token ID can be used to show the full object name.

### Privilege class

Any administrator can issue this command

### Syntax

►►—DISplay OBJname—*token\_ID*—◄◄

### Parameters

#### *token\_ID* (Required)

Specifies the ID reported in the [TSMOBJ:] tag, when an object name is too long to display.

### Example: Display the full object name of a token ID in a message

Assume the you receive the following message:

ANR9999D file.c(1999) Error handling file [TSMOBJ:1.1.649498] because of lack of server resources.

Display the full object name for the file referenced in the error message by specifying the token ID on the DISPLAY OBJNAME command.

display obj 1.1.649498

### Related commands

Table 152. Commands related to DISPLAY OBJNAME

| Command       | Description  |
|---------------|--|
| QUERY CONTENT | Displays information about files in a storage pool volume. |

---

### ENABLE commands

Use ENABLE commands to allow some types of operations by the server.

The following is a list of ENABLE commands for Tivoli Storage Manager:

- “ENABLE EVENTS (Enable server or client events for logging)” on page 417
- “ENABLE SESSIONS (Resume user activity on the server)” on page 420

## ENABLE EVENTS (Enable server or client events for logging)

Use this command to enable the processing of one or more events. If you specify a receiver that is not supported on any platform, or if you specify an invalid event or name, Tivoli Storage Manager issues an error message. However, any valid receivers, events, or names that you specified are still enabled.

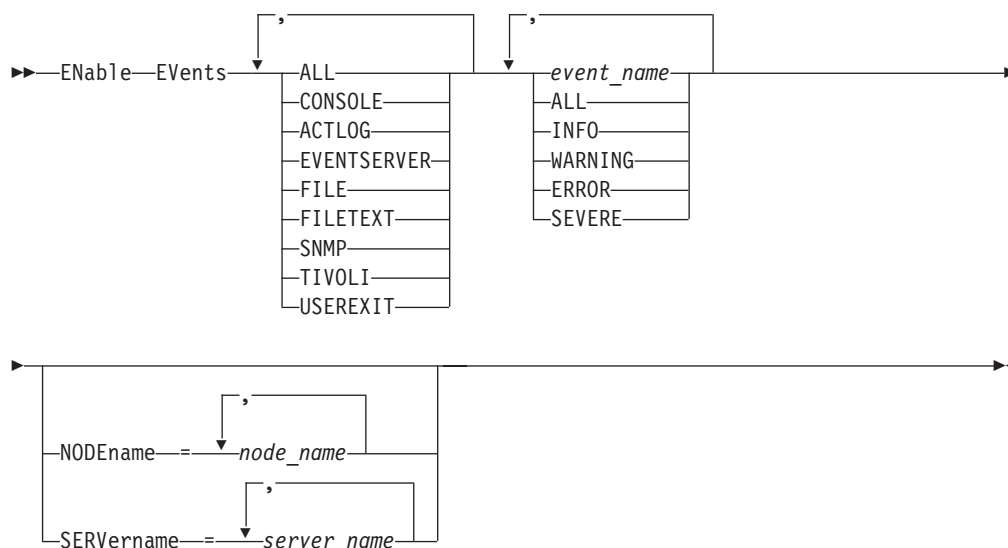
**Restriction:** Certain events, such as some messages issued during server start-up and shutdown, automatically go to the console. They do not go to other receivers even if they are enabled.

Administrative commands are returned to the command issuer and are only logged as numbered events. These numbered events are not logged to the system console, but are logged to other receivers, including administrative command-line sessions running in console mode.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *receivers* (Required)

Specifies one or more receivers for which to log enabled events. You can specify multiple receivers by separating them with commas and no intervening spaces. Valid values are:

#### **ALL**

All receivers.

#### **CONSOLE**

The standard server console as a receiver.

#### **ACTLOG**

The server activity log as a receiver.

## ENABLE EVENTS

### EVENTSERVER

The event server as a receiver.

### FILE

A user file as a receiver. Each logged event is a record in the file. The records are not easily readable by people.

### FILETEXT

A user file as a receiver. Each logged event is a fixed-size, readable line.

### SNMP

The simple network management protocol (SNMP) as a receiver.

### TIVOLI

The Tivoli Enterprise Console (TEC) as a receiver.

### USEREXIT

A user-written program as a receiver. The server writes information to the program.

### *events* (Required)

Specifies the type of events to be enabled. You can specify multiple events by separating them with commas and no intervening spaces. Possible values are:

#### ALL

All events.

#### *event\_name*

A four-digit message number preceded by ANR for a server event or ANE for a client event. Valid ranges are from ANR0001 to ANR9999 and from ANE4000 to ANE4999. Specify the NODENAME parameter if client events are to be enabled for matching nodes. Specify the SERVERNAME parameter if server events are to be enabled for matching servers.

For the TIVOLI event receiver, you can specify the following additional ranges for the Tivoli Storage Manager application clients:

| Tivoli Storage Manager application client     | Prefix | Range     |
|---|--------|-----------|
| Data Protection for Microsoft Exchange Server | ACN    | 3500–3649 |
| Data Protection for Lotus Domino              | ACD    | 5200–5299 |
| Data Protection for Oracle                    | ANS    | 500–599   |
| Data Protection for Informix                  | ANS    | 600–699   |
| Data Protection for Microsoft SQL Server      | ACO    | 3000–3999 |

**Restriction:** The application client must have enhanced Tivoli Event Console support enabled in order to route these messages to the Tivoli Event Console.

#### Tip:

- Specifying the ALL option enables these messages. However, the INFO, WARNING, ERROR, and SEVERE options have no effect on the messages.
- Because of the number of messages, you should not enable all messages from a node to be logged to the Tivoli Event Console.



*severity categories*

If the event list contains a severity category, all events of that severity are enabled for the specified nodes. The message types are:

**INFO**

Information messages (type of I) are enabled.

**WARNING**

Warning messages (type of W) are enabled.

**ERROR**

Error messages (type of E) are enabled.

**SEVERE**

Severe error messages (type of S) are enabled.

**NODENAME**

Specifies one or more client nodes for which events are enabled. You can use a wildcard character to specify all client nodes. You can specify NODENAME or SERVERNAME. If neither parameter is specified, events are enabled for the server running this command.

**SERVERNAME**

Specifies one or more servers for which events are to be enabled. You can use a wildcard character to specify all servers other than the server from which this command is issued. You can specify SERVERNAME or NODENAME. If neither parameter is specified, the events are enabled for the server running this command.

**Example: Enable specific categories of events**

Enable all ERROR and SEVERE client events to the USEREXIT receiver for the node BONZO.

```
enable events userexit error,severe nodename=bonzo
```

**Related commands**

*Table 153. Commands related to ENABLE EVENTS*

| Command            | Description   |
|--------------------|---|
| BEGIN EVENTLOGGING | Starts event logging to a specified receiver.   |
| DISABLE EVENTS     | Disables specific events for receivers.   |
| END EVENTLOGGING   | Ends event logging to a specified receiver.   |
| QUERY ENABLED      | Displays enabled or disabled events for a specific receiver.                            |
| QUERY EVENTRULES   | Displays information about rules for server and client events.                          |
| QUERY STATUS       | Displays the settings of server parameters, such as those selected by the SET commands. |

### ENABLE SESSIONS (Resume user activity on the server)

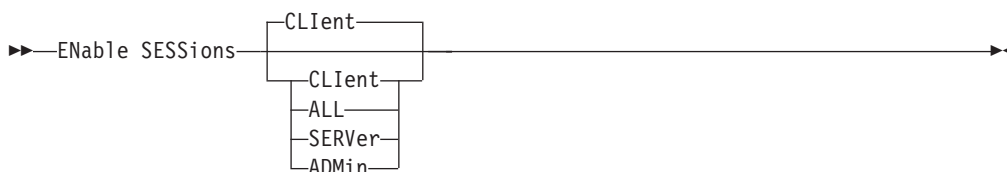
Use this command to allow activity to begin after issuing the `DISABLE SESSIONS` command. The processing of this command does not affect system processes, such as migration and reclamation.

Use the `QUERY STATUS` command to display the availability of the server.

#### Privilege class

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

#### Syntax



#### Parameters

Specifies the type of session to be enabled. This parameter is optional. The default value is `CLIENT`. Possible values are:

##### **CLient**

Enables only backup and archive client sessions.

##### **ALL**

Enables all session types.

##### **SERVer**

Enables only server-to-server sessions.

##### **ADMin**

Enables only administrative sessions.

#### Example: Resume client node activity on the server

Resume normal operation, permitting client nodes to access the server.

```
enable sessions
```

#### Example: Resume all activity on the server

Resume normal operation, permitting all sessions to access the server.

```
enable sessions all
```

#### Related commands

Table 154. Commands related to `ENABLE SESSIONS`

| Command                     | Description                              |
|-----------------------------|--|
| <code>ACCEPT DATE</code>    | Accepts the current date on the server.  |
| <code>CANCEL SESSION</code> | Cancels active sessions with the server. |

*Table 154. Commands related to ENABLE SESSIONS (continued)*

| Command          | Description  |
|------------------|--|
| DISABLE SESSIONS | Prevents new sessions from accessing IBM Tivoli Storage Manager but permits existing sessions to continue. |
| QUERY SESSION    | Displays information about all active administrator and client sessions with IBM Tivoli Storage Manager.   |
| QUERY STATUS     | Displays the settings of server parameters, such as those selected by the SET commands.                    |

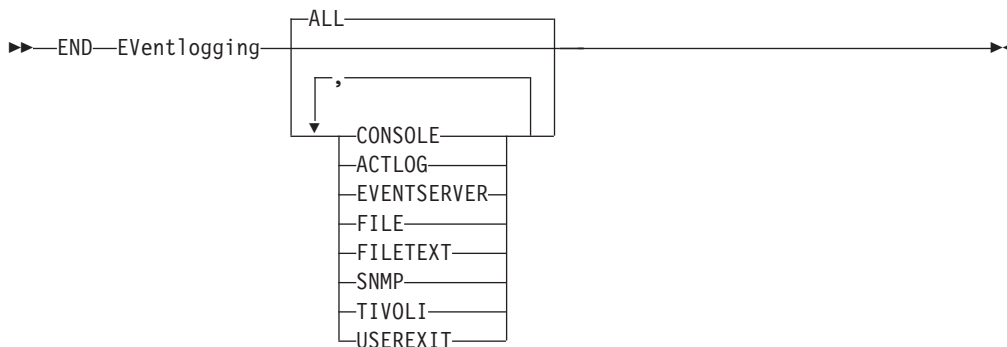
## END EVENTLOGGING (Stop logging events)

Use this command to stop logging events to an active receiver.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

Specify a type of receiver. You can specify multiple receivers by separating them with commas and no intervening spaces. This is an optional parameter. The default is ALL. If you specify ALL or no receiver, logging ends for all receivers.

#### ALL

Specifies all receivers.

#### CONSOLE

Specifies the server console as a receiver.

#### ACTLOG

Specifies the Tivoli Storage Manager activity log as a receiver. Logging can be stopped only for client events.

#### EVENTSERVER

Specifies the event server as a receiver.

#### FILE

Specifies a user file as a receiver. Each logged event is a record in the file and a person cannot read each logged event easily.

#### FILETEXT

Specifies a user file as a receiver. Each logged event is a fixed-size, readable line.

#### SNMP

Specifies the simple network management protocol (SNMP) as a receiver.

#### TIVOLI

Specifies the Tivoli Management Environment (TME) as a receiver.

#### USEREXIT

Specifies a user-written routine to which Tivoli Storage Manager writes information as a receiver.

### Example: Stop logging events

End logging of events to the user exit.

```
end eventlogging userexit
```

### Related commands

*Table 155. Commands related to END EVENTLOGGING*

| Command            | Description   |
|--------------------|---|
| BEGIN EVENTLOGGING | Starts event logging to a specified receiver.   |
| DISABLE EVENTS     | Disables specific events for receivers.   |
| ENABLE EVENTS      | Enables specific events for receivers.  |
| QUERY ENABLED      | Displays enabled or disabled events for a specific receiver.                            |
| QUERY EVENTRULES   | Displays information about rules for server and client events.                          |
| QUERY STATUS       | Displays the settings of server parameters, such as those selected by the SET commands. |

## EXPIRE INVENTORY (Manually start inventory expiration processing)

Use this command to manually start inventory expiration processing. The inventory expiration process removes client backup and archive file copies from server storage based on policy specified in the backup and archive copy groups of the management classes to which the files are bound.

When you have the disaster recovery manager function for your Tivoli Storage Manager server, the inventory expiration process also removes eligible virtual volumes that are used by the following processes:

- Database backups of type BACKUPFULL, BACKUPINCR, and DBSNAPSHOT. The SET DRMDBBACKUPEXPIREDAYS command controls when these volumes are eligible for expiration.
- Recovery plan files of type RPFILE and RPFSNAPSHOT. The SET DRMRPFEXPIREDAYS command controls when these volumes are eligible for expiration.

The inventory expiration process that runs during server initialization does not remove these virtual volumes.

Only one expiration process is allowed at any time, but this process can be distributed among threads (maximum 10). If an expiration process is currently running, you cannot start another process.

You can set up automatic expiration processing with the EXPINTERVAL server option. If you set the EXPINTERVAL to 0, the server does not run expiration automatically, and you must issue this command to start expiration processing.

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.

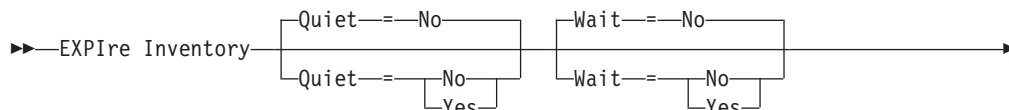
If this command is applied to a WORM (write once, read many) volume, the volume will return to scratch if it has space remaining in which data can be written. (Note that data on WORM volumes, including deleted and expired data, cannot be overwritten. Therefore, data can only be written in space that does not contain current, deleted, or expired data.) If a WORM volume does not have any space available in which data can be written, it will remain private. To remove the volume from the library, you must use the CHECKOUT LIBVOLUME command.

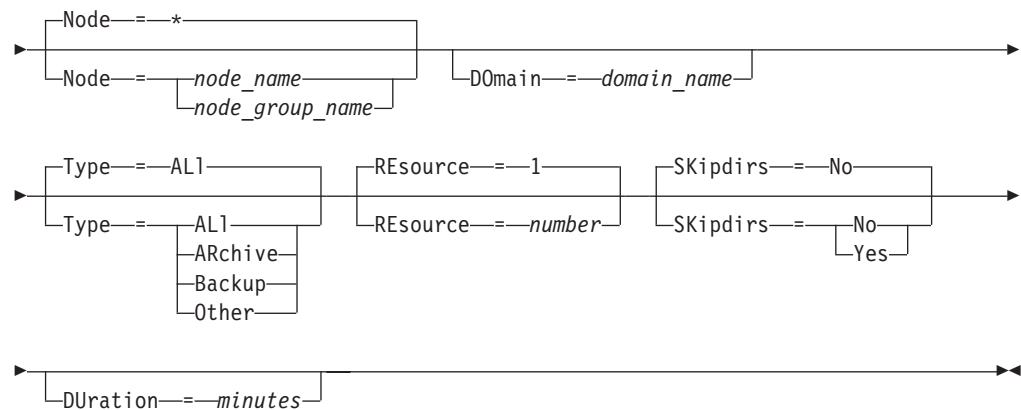
Files will not be completely deleted from server storage using client delete operations, so you should consider running the EXPIRE INVENTORY command. See the *Backup-Archive Clients Installation and User's Guide* for more information about client delete operations.

### Privilege class

To issue this command, you must have system privilege.

### Syntax





## Parameters

### Quiet

Specifies whether the server suppresses detailed messages about policy changes during the expiration processing. This parameter is optional. The default is NO. Possible values are:

#### No

Specifies that the server sends detailed informational messages.

#### Yes

Specifies that the server sends only summary messages. The server issues 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.

You can also specify the EXPQUIET option in the server options file to automatically determine if expiration processing is performed with summary messages.

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

The server displays messages that are created from the background process either in the activity log or the server console, depending on where messages are logged.

#### Yes

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

### Node

Specifies the name of the client nodes or node groups whose data is to be processed. To specify multiple node and node group names, separate the

## EXPIRE INVENTORY

names with commas and no intervening spaces. Node names can contain wildcard characters, but node group names cannot.

You can specify either **NODE** or **DOMAIN**. If you specify both, only those nodes that match the criteria for both specified command options are processed. If you do not specify either **NODE** or **DOMAIN** with a value, data for all nodes is processed.

### **Domain**

Specifies that only data for client nodes assigned to the specified domain is to be processed. You can specify either **NODE** or **DOMAIN**. If you specify both, only those nodes that match the criteria for both specified command options are processed. If you do not specify either **NODE** or **DOMAIN** with a value, data for all nodes is processed.

### **Type**

Specifies the type of data to be processed. The default value is **ALL**. Possible values are:

#### **ALL**

Process all types of data that are eligible for expiration

#### **ARchive**

Process only client archive data

#### **Backup**

Process only client backup data

#### **Other**

Process only items for disaster recovery manager functions, such as recovery plan files and obsolete database backups

### **REsource**

Specifies the number of threads that can run in parallel. Specify a number from 1 to 10.

The resources represent parallel work by the server within the single expiration process (expiration still runs as a single process). For example, if you specify **NODE=X,Y,Z** and **RESOURCE=3** (or greater), then expiration processing for the three client nodes X, Y, and Z runs in parallel.

If you specify **NODE=X,Y,Z** and **RESOURCE=5**, then expiration processing for the three client nodes runs in parallel, and the extra two resources are ignored.

### **DUration**

Specifies the maximum number of minutes for the expiration process to run. The process stops when the specified number of minutes pass or when all eligible expired objects are deleted, whichever comes first. You can specify a number from 1 to 600. This parameter is optional. If this parameter is not specified, the duration of the expiration process is not limited by time.

### **Example: Run inventory expiration processing for a specific time period**

Run the expiration process for two hours.

```
expire inventory duration=120
```



**Example: Run inventory expiration processing for backup data for two client nodes**

Run inventory expiration processing for the backup data for two client nodes, CHARLIE and ROBBIE. Let the server run expiration processing until completed.

```
expire inventory node=charlie,robbie resource=2 type=backup
```

**Related commands**

*Table 156. Commands related to EXPIRE INVENTORY*

| Command        | Description                                      |
|----------------|--|
| AUDIT LICENSES | Checks for compliance with defined licenses.     |
| CANCEL PROCESS | Cancels a background server process.             |
| QUERY PROCESS  | Displays information about background processes. |

---

## EXPORT commands

Use the EXPORT commands to copy information from a Tivoli Storage Manager server to sequential removable media.

The following is a list of EXPORT commands for Tivoli Storage Manager:

- “EXPORT ADMIN (Export administrator information)” on page 429
- “EXPORT NODE (Export client node information)” on page 435
- “EXPORT POLICY (Export policy information)” on page 453
- “EXPORT SERVER (Export server information)” on page 459

## EXPORT ADMIN (Export administrator information)

Use this command to export administrator and authority definitions from a server. You can export the information to sequential media for later importing to another server, or you can export the information directly to another server for immediate import.

### Important:

- If target and source server levels are not compatible, the operation might not work. See the *Administrator's Guide* for server compatibility requirements.
- Exporting data to a CENTERA device class is not supported. However, files stored in CENTERA storage pools can be exported.

Tivoli Storage Manager exports administrator information such as:

- Administrator name, password, and contact information
- Administrative privilege classes the administrator has been granted
- Whether the administrator ID is locked from server access

You can use the QUERY ACTLOG command to view the status of the export operation. You can also view this information from the server console.

This command generates a background process that can be cancelled with the CANCEL PROCESS command. If you export information to sequential media and the background process is cancelled, the sequential media holding the exported data are incomplete and should not be used for importing data. If a server-to-server export background process is cancelled, a partial import may result. Evaluate any imported data on the target server to determine if you want to keep or delete the imported data. Review the import messages for details. To display information on background processes, use the QUERY PROCESS command.

The EXPORT ADMIN command takes two forms: export directly to another server on the network, or export to sequential media. The syntax and parameters for each form are defined separately.

Table 157. Commands related to EXPORT ADMIN

| Command        | Description  |
|----------------|--|
| CANCEL PROCESS | Cancels a background server process.                     |
| EXPORT NODE    | Copies client node information to external media.        |
| EXPORT POLICY  | Copies policy information to external media.             |
| EXPORT SERVER  | Copies all or part of the server to external media.      |
| IMPORT ADMIN   | Restores administrative information from external media. |
| QUERY ACTLOG   | Displays messages from the server activity log.          |
| QUERY PROCESS  | Displays information about background processes.         |

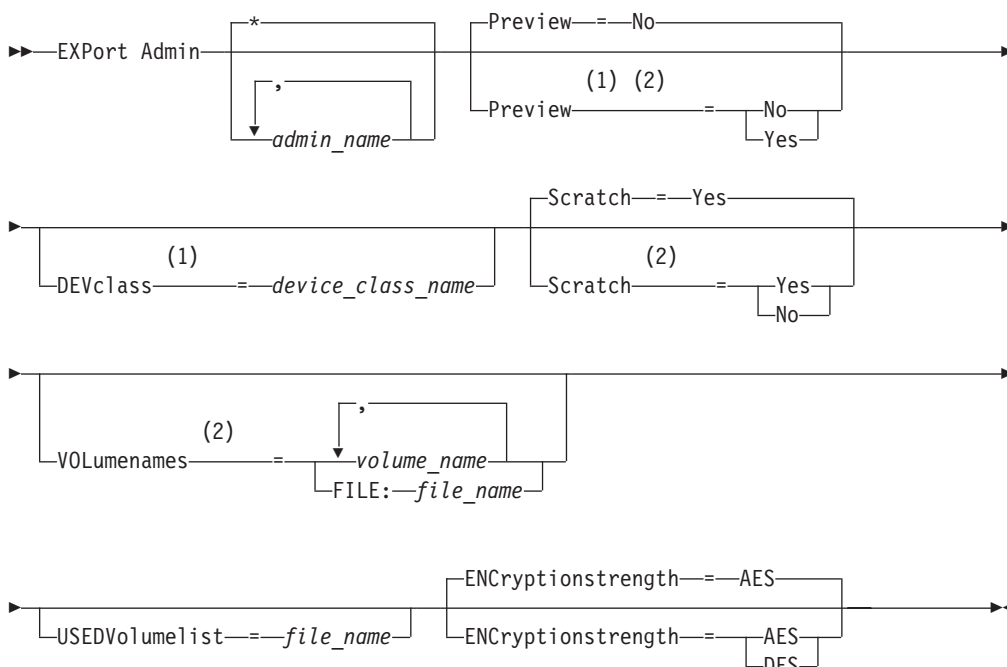
### EXPORT ADMIN (Export administrator definitions to sequential media)

You can export administrator and authority definitions from a server to sequential media for later importing to another server.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax



#### Notes:

- 1 If `PREVIEW=NO`, a device class must be specified.
- 2 If `PREVIEW=NO` and `SCRATCH=NO`, one or more volumes must be specified.

#### Parameters

##### *admin\_name*

Specifies the administrators for which information is to be exported. This parameter is optional. The default is all administrators.

Separate the items in the list by commas, with no intervening spaces. You can use wildcard characters to specify names.

##### Preview

Specifies whether to preview the results of the export operation, without 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. This parameter is optional. The default value is `NO`. The values are:

##### No

Specifies that the administrator information is to be exported. If you specify this value, you must specify a device class.

**Yes**

Specifies that the operation will be previewed but not performed. Information is reported to the server console and the activity log. If you specify this value, you do not need to specify a device class.

**DEVclass**

Specifies the device class to which export data is to be written. This parameter is required if you specify PREVIEW=NO.

You cannot specify the DISK, NAS, or CENTERA device classes.

If all drives for the device class are busy when the export runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

**Tip:** You can export data to a storage pool on another server by specifying a device class whose device type is SERVER. For details about storing data on another server, see the *Administrator's Guide*.

**Scratch**

Specifies whether scratch volumes can be used. The default value is YES. Possible values are:

**Yes**

Specifies that scratch volumes can be used for export. If you also specify a list of volumes, scratch volumes are used only if there is not enough space on the volumes specified.

**No**

Specifies that scratch volumes cannot be used for export. To determine how many volumes you might need, you can run the command specifying PREVIEW=YES.

**VOLumenames**

Specifies the volumes to be used to contain exported data. This parameter is optional, unless you specify SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, scratch volumes are used.

Possible values are:

*volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify  |
|-----------------|--|
| Tape            | 1–6 alphanumeric characters.   |
| FILE            | Any fully qualified file name string. For example:<br><br>/imdata/mt1. |
| OPTICAL         | 1–32 alphanumeric characters.  |
| REMOVABLEFILE   | 1–6 alphanumeric characters.   |

## EXPORT ADMIN—to sequential media

| For this device | Specify                        |
|-----------------|--------------------------------|
| SERVER          | 1–250 alphanumeric characters. |

### USEDVolumelist

Specifies the file where a list of volumes used in the export operation will be stored. This parameter is optional.

This file can be used as input to the import operation. This file will contain comment lines with the date and time the export was done, and the command issued to create the export.

**Attention:** If you specify an existing file, the file is overwritten.

### ENCryptionstrength

Indicates which algorithm to use to encrypt passwords when exporting administrative and node records. This parameter is optional. The default value is AES. If you are exporting to a server that does not support AES, specify DES. Possible values are:

#### AES

Specifies the Advanced Encryption Standard.

#### DES

Specifies the Data Encryption Standard.

### Example: Export administrator definitions to tape volumes

From the server, export the information for all defined administrators to tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class. The number and type of objects exported are reported to the system console and in the activity log. Issue the command:

```
export admin devclass=menu1  
volumenames=tape01,tape02,tape03
```

### Example: Export administrator definitions to tape volumes listed in a file

From the server, export the information for all defined administrators to tape volumes that are listed in the following file:

TAPEVOL

This file contains the following lines:

```
TAPE01  
TAPE02  
TAPE03
```

Specify that these tape volumes be used by a device assigned to the MENU1 device class. Issue the command:

```
export admin devclass=menu1 volumenames=file:tapevol
```

The number and type of objects exported are reported to the system console and in the activity log.

## EXPORT ADMIN (Export administrator information directly to another server)

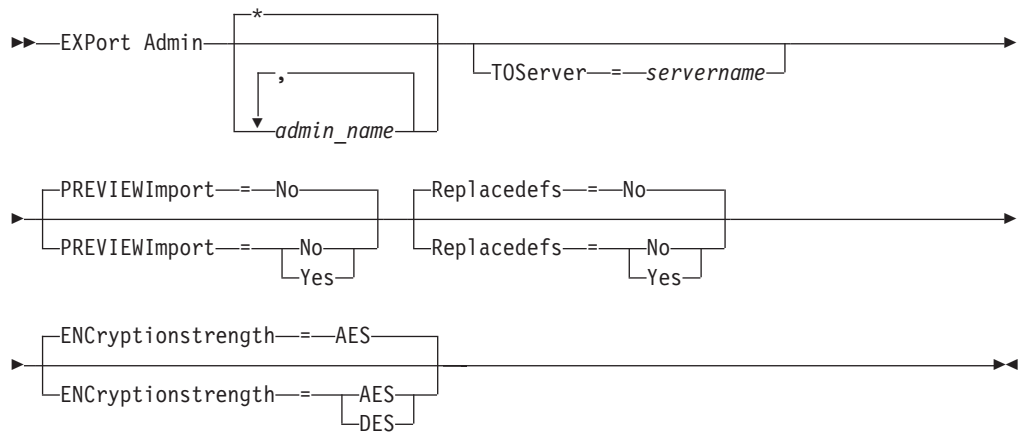
Use this command to export administrator and authority definitions directly to another server on the network. This results in an immediate import on the target server.

You can issue a QUERY PROCESS command from the target server to monitor the progress of the import operation.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *admin\_name*

Specifies the administrators for which information is to be exported. This parameter is optional. The default is all administrators.

Separate the items in the list by commas, with no intervening spaces. You can use wildcard characters to specify names.

#### **TOServer**

Specifies the name of a server to which the export data will be sent directly over the network for immediate import.

**Important:** The target server must be defined on the originating server with the DEFINE SERVER command. The administrator that issues the export command must be defined with the same administrator name and password and have system authority on the target server.

When you specify TOSERVER, you cannot specify the DEVCLASS, VOLUMENAMES, and SCRATCH, USEDVOLUMELIST, and PREVIEW parameters.

#### **PREVIEWImport**

Specifies whether to view how much data will be transferred, without actually moving any data. This information can be used to determine how much storage pool space is required on the target server. The default is NO.

Valid values are:

## EXPORT ADMIN—directly to another server

### Yes

Specifies that you want to preview the results of the import operation on the target server, without importing the data. Information is reported to the server console and the activity log.

### No

Specifies that you want the data to be imported on the target server without previewing the results.

### Replacedefs

Specifies whether to replace definitions (not file data) on the server. The default is NO.

Valid values are:

### Yes

Specifies that definitions are replaced on the server if definitions having the same name as those being imported already exist on the target server.

### No

Specifies that imported definitions are skipped if their names conflict with definitions that are already defined on the target server.

### ENCryptionstrength

Indicates which algorithm to use to encrypt passwords when exporting administrative and node records. This parameter is optional. The default value is AES. If you are exporting to a server that does not support AES, specify DES. Possible values are:

### AES

Specifies the Advanced Encryption Standard.

### DES

Specifies the Data Encryption Standard.

## Example: Export administrator definitions to a target server

Export all the administrator definitions to the target server defined as OTHERSERVER. Preview the import operations on the target server. Issue the command:

```
export admin * toserver=otherserver previewimport=yes
```

From the target server, OTHERSERVER, you can view the import operations by issuing the command:

```
query process
```



## EXPORT NODE (Export client node information)

Use this command to export client node definitions or file data to sequential media or directly to another server for immediate import.

You can export data from a server with retention protection enabled, but it is not retention protected when imported on another server.

You cannot export nodes of type NAS; export processing excludes these nodes.

The following information is included in each client node definition:

- User ID, password, and contact information.
- Name of the client's assigned policy domain.
- File compression status.
- Whether the user has the authority to delete backed-up or archived files from server storage.
- Whether the client node ID is locked from server access.

Optionally, you can also export the following items:

- File space definitions.
- Backed-up, archived, and files that were migrated by a Tivoli Storage Manager for Space Management client.
- Access authorization information pertaining to the file spaces exported.
- Archive data that is in deletion hold status (the hold status will be preserved).  
When the archive data is imported, it will remain in deletion hold.

The EXPORT NODE command generates a background process that can be cancelled with the CANCEL PROCESS command. If you are exporting node information to sequential media and the background process is cancelled, the sequential media that is holding the exported data are incomplete and should not be used for importing data. If a server-to-server export background process is cancelled, a partial import may result. Evaluate any imported data on the target server to determine if you want to keep or delete the imported data. Review the import messages for details. To display information on background processes, issue the QUERY PROCESS command.

To display information on any running and suspended server-to-server export operations, issue the QUERY EXPORT command. The QUERY EXPORT command only displays information for exports that are, or can be, suspended. Export operations that can be suspended, and subsequently restarted, are those server-to-server exports whose FILEDATA has a value other than NONE. You can issue the QUERY ACTLOG command to view the status of the export operation.

### Restriction:

1. If target and source server levels are not compatible, the operation might not work. See the *Administrator's Guide* for server compatibility requirements.
2. Exporting data to a CENTERA device class is not supported. However, files stored in CENTERA storage pools can be exported.

Because of unpredictable results, do not run expiration, migration, backup, or archive when issuing the EXPORT NODE command. See the *Administrator's Guide* for more information.

## EXPORT NODE

For a server that has clients with support for Unicode, you might need to have the server convert the file space name that you enter, or use one of the following parameters:

- **FSID**
- **UNIFILESPACE**

The EXPORT NODE command takes two forms: export directly to another server on the network, or export to sequential media. The syntax and parameters for each form are defined separately.

*Table 158. Commands related to EXPORT NODE*

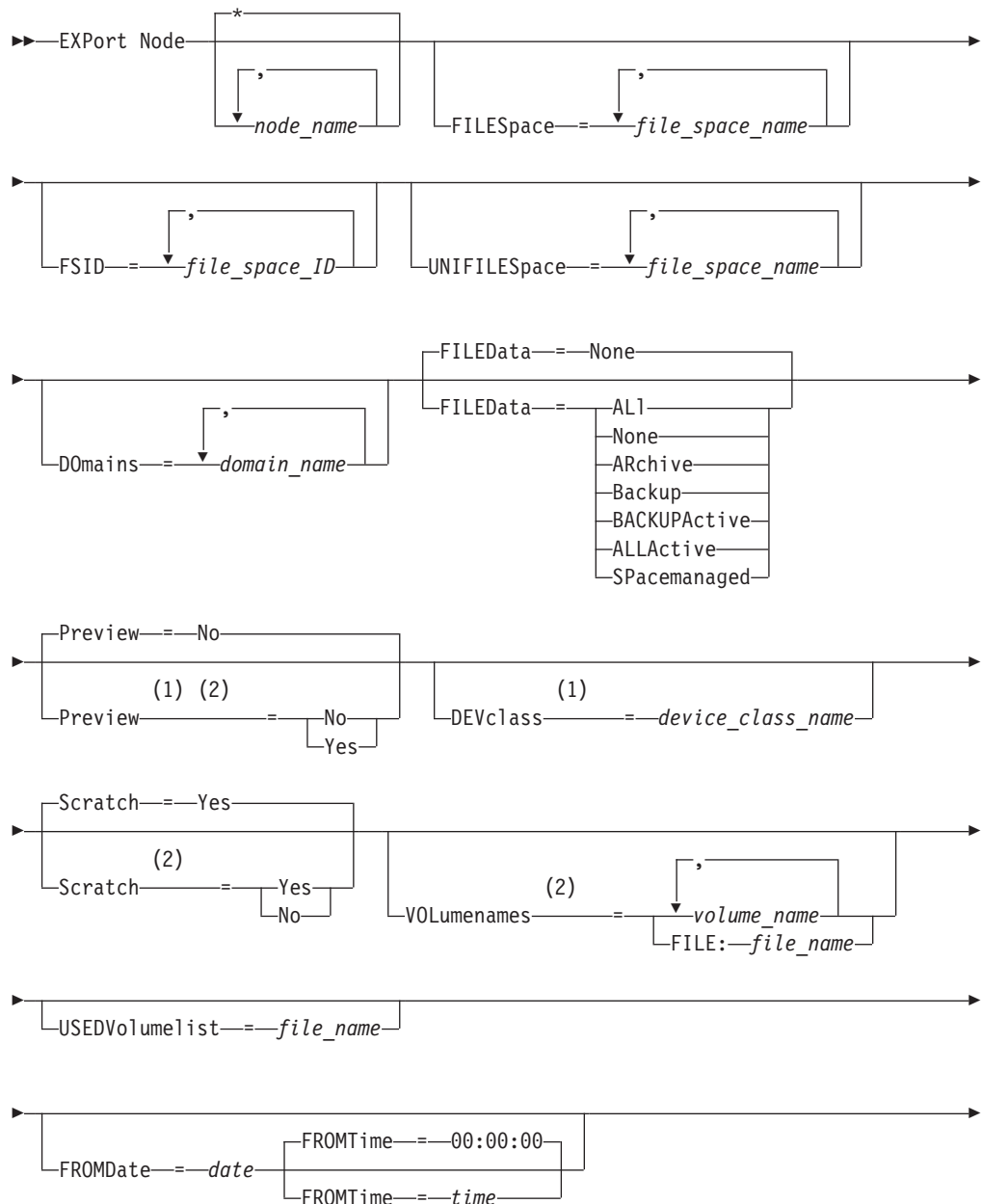
| Command           | Description   |
|-------------------|---|
| CANCEL EXPORT     | Deletes a suspended export operation                                    |
| CANCEL PROCESS    | Cancels a background server process.                                    |
| COPY ACTIVATEDATA | Copies active backup data.  |
| EXPORT ADMIN      | Copies administrative information to external media.                    |
| EXPORT POLICY     | Copies policy information to external media.                            |
| EXPORT SERVER     | Copies all or part of the server to external media.                     |
| IMPORT NODE       | Restores client node information from external media.                   |
| QUERY ACTLOG      | Displays messages from the server activity log.                         |
| QUERY EXPORT      | Displays the export operations that are currently running or suspended. |
| QUERY PROCESS     | Displays information about background processes.                        |
| RESTART EXPORT    | Restarts a suspended export operation.                                  |
| SUSPEND EXPORT    | Suspends a running export operation.                                    |

**EXPORT NODE (Export node definitions to sequential media)**

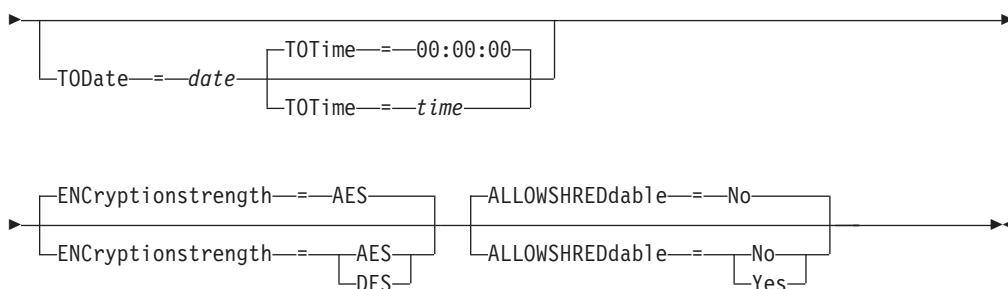
You can export node definitions or file data from a server to sequential media for later importing to another server.

**Privilege class**

To issue this command, you must have system privilege.

**Syntax**

## EXPORT NODE-to sequential media



**Notes:**

- 1 If PREVIEW=NO, a device class must be specified.
- 2 If PREVIEW=NO and SCRATCH=NO, one or more volumes must be specified.

## Parameters

node name

Specifies the client node names for which information is to be exported. This parameter is optional. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. For each node entered, all file spaces in the file space, FSID, and Unicode enabled lists will be searched.

**Restriction:** If you use wildcard characters to specify a pattern for node names, the server will not report the node names or patterns that do not match any entries in the database. Check the summary statistics in the activity log to verify that the server exported all intended nodes.

## FILESpace

Specifies the file spaces for which data is to be exported. This parameter is optional. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify a name.

## FSID

Specifies the file spaces by using their file space IDs (FSIDs). The server uses the FSIDs to find the file spaces to export. To find the FSID for a file space, use the `QUERY FILESPACE` command. Separate multiple file space IDs with commas and no intervening spaces. This parameter is optional.

## UNIFILESpace

Specifies the file spaces that are known to the server to be Unicode enabled. The server converts the names you enter from the server code page to the UTF-8 code page to find the file spaces to export. The success of the conversion depends on the actual characters in the name and the server's code page. Separate multiple names with commas and no intervening spaces. This parameter is optional.

## DOmains

Specifies the policy domains from which nodes should be exported. This parameter is optional. Separate multiple names with commas and no intervening spaces. If you specify domains, a node is exported only if it belongs to one of the specified domains. You can use wildcard characters to specify a name.

**FILEData**

Specifies the type of files that should be exported for all nodes being exported to the server. This parameter is optional. The default value is NONE.

If you are exporting to sequential media: 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, two drives are needed to export node information. The mount limit for the device class must be at least 2.

**Important:** If client nodes registered as TYPE=SERVER are being exported, specify ALL, ARCHIVE, or ALLACTIVE.

The following descriptions mention *active* and *inactive* backup file versions. An active backup file version is the most recent backup version for a file that still exists on the client workstation. All other backup file versions are called inactive copies. The values are:

**ALL**

The server exports all backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client.

**None**

The server does not export files, only node definitions.

**Archive**

The server exports only archived files.

**Backup**

The server exports only backup versions, whether active or inactive.

**BACKUPActive**

The server exports only active backup versions. These active backup versions are the active versions in the Tivoli Storage Manager database at the time that the EXPORT command is issued.

**ALLActive**

The server exports all active backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client. The active backup versions are the active versions in the Tivoli Storage Manager database at the time that the EXPORT command is issued.

**SPacemanaged**

The server exports only files that were migrated by a Tivoli Storage Manager for Space Management client.

**Preview**

Specifies whether to preview the results of the export operation, without exporting information. You can use this parameter to preview how many bytes of data would be transferred, allowing you to determine how many volumes will be required. This parameter is optional. The default value is NO. The values are:

**No**

Specifies that the node information is to be exported. If you specify this value, you must also specify a device class.

**Yes**

Specifies that the operation will be previewed but not performed. Information is reported to the server console and the activity log. If you specify this value, you do not need to specify a device class.

## EXPORT NODE—to sequential media

### DEVclass

Specifies the device class to which export data is to be written. This parameter is required if you specify PREVIEW=NO.

You cannot specify the DISK, NAS, or CENTERA device classes.

If all drives for the device class are busy when the export runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

**Tip:** You can export data to a storage pool on another server by specifying a device class whose device type is SERVER. For details about storing data on another server, see the *Administrator's Guide*.

### Scratch

Specifies whether scratch volumes can be used. The default value is YES. Possible values are:

#### Yes

Specifies that scratch volumes can be used for export. If you also specify a list of volumes, scratch volumes are used only if there is not enough space on the volumes specified.

#### No

Specifies that scratch volumes cannot be used for export. To determine how many volumes you might need, you can run the command specifying PREVIEW=YES.

### VOLumenames

Specifies the volumes to be used to contain exported data. This parameter is optional, unless you specify SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, scratch volumes are used.

Possible values are:

#### *volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

#### **FILE:***file\_name*

Specifies the name of a file that contains a list of volumes. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify  |
|-----------------|--|
| Tape            | 1-6 alphanumeric characters.   |
| FILE            | Any fully qualified file name string. For example:<br><br>/imdata/mt1. |
| OPTICAL         | 1-32 alphanumeric characters.  |
| REMOVABLEFILE   | 1-6 alphanumeric characters.   |
| SERVER          | 1-250 alphanumeric characters.   |

### USEDVolumelist

Specifies the file where a list of volumes used in the export operation will be stored. This parameter is optional.

This file can be used as input to the import operation. This file will contain comment lines with the date and time the export was done, and the command issued to create the export.

**Attention:** If you specify an existing file, the file is overwritten.

#### FROMDate

Specifies the earliest date for which objects to be exported were stored on the server. Objects that were stored on the server earlier than the specified date will not be exported. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

You can specify the date by using one of the values below:

| Value                      | Description  | Example         |
|----------------------------|--|-----------------|
| <i>MM/DD/YYYY</i>          | A specific date  | 09/15/1998      |
| <b>TODAY</b>               | The current date   | TODAY           |
| <b>TODAY-days or -days</b> | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

If this parameter is not specified, Tivoli Storage Manager exports all objects stored before the TODATE parameter and as qualified by the FILEDATA parameter. If no TODATE parameter is specified, then all data as qualified by the FILEDATA parameter is exported.

#### TODate

Specifies the latest date for objects to be exported from the server. Objects stored on the server on a date later than the TODATE value are not exported. TODATE only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TODATE parameter when the FILEDATA parameter is set to NONE.

**Important:** If a TODATE parameter is specified without a TOTIME parameter, the server exports all objects inserted on or before the day specified by the TODATE parameter.

You can specify the date by using one of the values below:

| Value                      | Description  | Example         |
|----------------------------|--|-----------------|
| <i>MM/DD/YYYY</i>          | A specific date  | 10/15/2006      |
| <b>TODAY</b>               | The current date   | TODAY           |
| <b>TODAY-days or -days</b> | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

**Important:** If you specified the FROMDATE parameter, the value of TODATE must be later than or equal to the FROMDATE value. If the TODATE and FROMDATE are equal, then the TOTIME parameter must be later than the FROMTIME parameter.

#### FROMTime

Specifies the earliest time for which objects to be exported were stored on the

## EXPORT NODE—to sequential media

server. When you specify FROMTIME, you must also use the FROMDATE parameter. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Objects that were stored on the server before the specified time and date will not be exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

You can specify the time by using one of the values below:

| Value                                | Description   | Example  |
|--------------------------------------|---|--|
| <i>HH:MM:SS</i>                      | A specific time   | 10:30:08   |
| <b>NOW</b>                           | The current time  | NOW  |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes specified. The FROMTIME+ can only be used with a FROMDATE before today. | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW+02:00 or FROMTIME=+02:00, the export operation only contains files that were put on the server after 7:00 on the FROMDATE that you specify. |
| <b>NOW-HH:MM</b><br><b>or -HH:MM</b> | The current time minus hours and minutes specified  | NOW-02:00 <b>or</b> -02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW-02:00 or FROMTIME=-2:00, the export includes files that were put on the server after 3:00.  |

### TOTime

Specifies the latest time that objects to be exported were stored on the server. You must specify the TODATE parameter in order to use the TOTIME parameter. TOTIME only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TOTIME parameter if the FILEDATA parameter is set to NONE.

The default value for this parameter, when used with the TODATE parameter, is midnight minus one second (23:59:59).

**Important:** The value of the TOTIME and TODATE parameters must be later than the FROMDATE and the FROMTIME value.

You can specify the time by using one of the values below:

| Value                                | Description  | Example  |
|--------------------------------------|--|--|
| <i>HH:MM:SS</i>                      | A specific time                                    | 10:30:08   |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes specified. | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW+02:00, the export includes files that were stored from 01:00 until 07:00. |



| Value                 | Description   | Example   |
|-----------------------|---|---|
| NOW-HH:MM<br>or-HH:MM | The current time minus hours and minutes specified. | NOW-02:00 or -02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW-02:00, the export includes files that were stored from 01:00 until 03:00. |

**ENCryptionstrength**

Indicates which algorithm to use to encrypt passwords when exporting administrative and node records. This parameter is optional. The default value is AES. If you are exporting to a server that does not support AES, specify DES. Possible values are:

**AES**

Specifies the Advanced Encryption Standard.

**DES**

Specifies the Data Encryption Standard.

**ALLOWSHREDdable**

Specifies whether data from a storage pool that enforces shredding is exported. This parameter is optional. The default value is NO. Possible values are:

**No**

Specifies that data is not exported from a storage pool that enforces shredding.

**Yes**

Specifies that data can be exported from a storage pool that enforces shredding. The data on the export media will not be shredded.

**Example: Export client node information to specific tape volumes**

From the server, export client node information to tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be used by a device assigned to the MENU1 device class.

```
export node devclass=menu1 volumenames=tape01,tape02,tape03
```

**Example: Export client node information using the FSID**

From the server, use the FSID to export active backup versions of file data for client node JOE to tape volume TAPE01. To determine the FSID, first issue a QUERY FILESPACE command.

1. To determine the FSID, issue a QUERY FILESPACE command.

```
query filespace joe
```

| Node Name | Filespace Name | FSID | Platform | Filespace Type | Is Filespace Unicode? | Capacity (MB) | Pct Util |
|-----------|----------------|------|----------|----------------|-----------------------|---------------|----------|
| JOE       | \\joe\c\$      | 1    | WinNT    | NTFS           | Yes                   | 2,502.3       | 75.2     |
| JOE       | \\joe\d\$      | 2    | WinNT    | NTFS           | Yes                   | 6,173.4       | 59.6     |

2. Export the active backup versions of file data and specify that the tape volume be used by a device assigned to the MENU1 device class.

```
export node joe fsid=1,2 filedata=backupactive devclass=menu1
volumenames=tape01
```

## EXPORT NODE—to sequential media

### Example: Export client node information to tape volumes listed in a file

From the server, export client node information to tape volumes that are listed in the following file:

```
TAPEVOL
```

The file contains the following lines:

```
TAPE01  
TAPE02  
TAPE03
```

Specify that the tape volumes be used by a device assigned to the MENU1 device class. Issue the following command:

```
export node devclass=menu1 volumenames=file:tapevol
```

## EXPORT NODE (Export node definitions or file data directly to another server)

Use this command to export client node definitions or file data directly to another server for immediate import.

**Important:** You cannot export nodes of type NAS. Export processing will exclude these nodes.

You can suspend and subsequently restart a server-to-server export operation that has a FILEDATA value other than NONE. The server saves the state and status of the export operation so that it may be restarted from the point at which the operation failed or was suspended. The export operation can be restarted at a later date by issuing the RESTART EXPORT command.

**Important:** An export operation is suspended when any of the following conditions are detected:

- A SUSPEND EXPORT command is issued for the running export operation
- Segment preemption - the file being read for export is deleted by some other process
- Communication errors on a server-to-server export
- No available mount points
- Necessary volumes are unavailable
- I/O errors encountered

Issue the QUERY EXPORT command to display information on any running and suspended export operations.

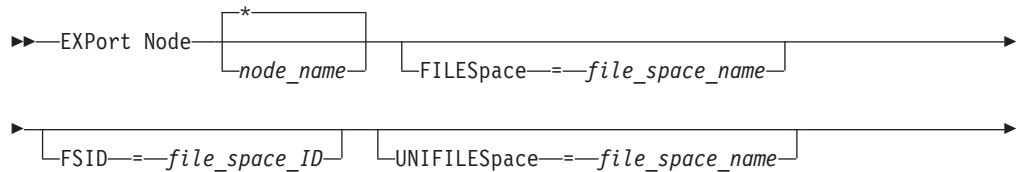
The export operation cannot be restarted if the export operation fails prior to transmitting the eligible node and file space definitions to the target server. You must reenter the command to begin a new export operation.

You can issue a QUERY PROCESS command from the target server to monitor the progress of the import operation. Issue the QUERY EXPORT command to list all restartable server-to-server export operations.

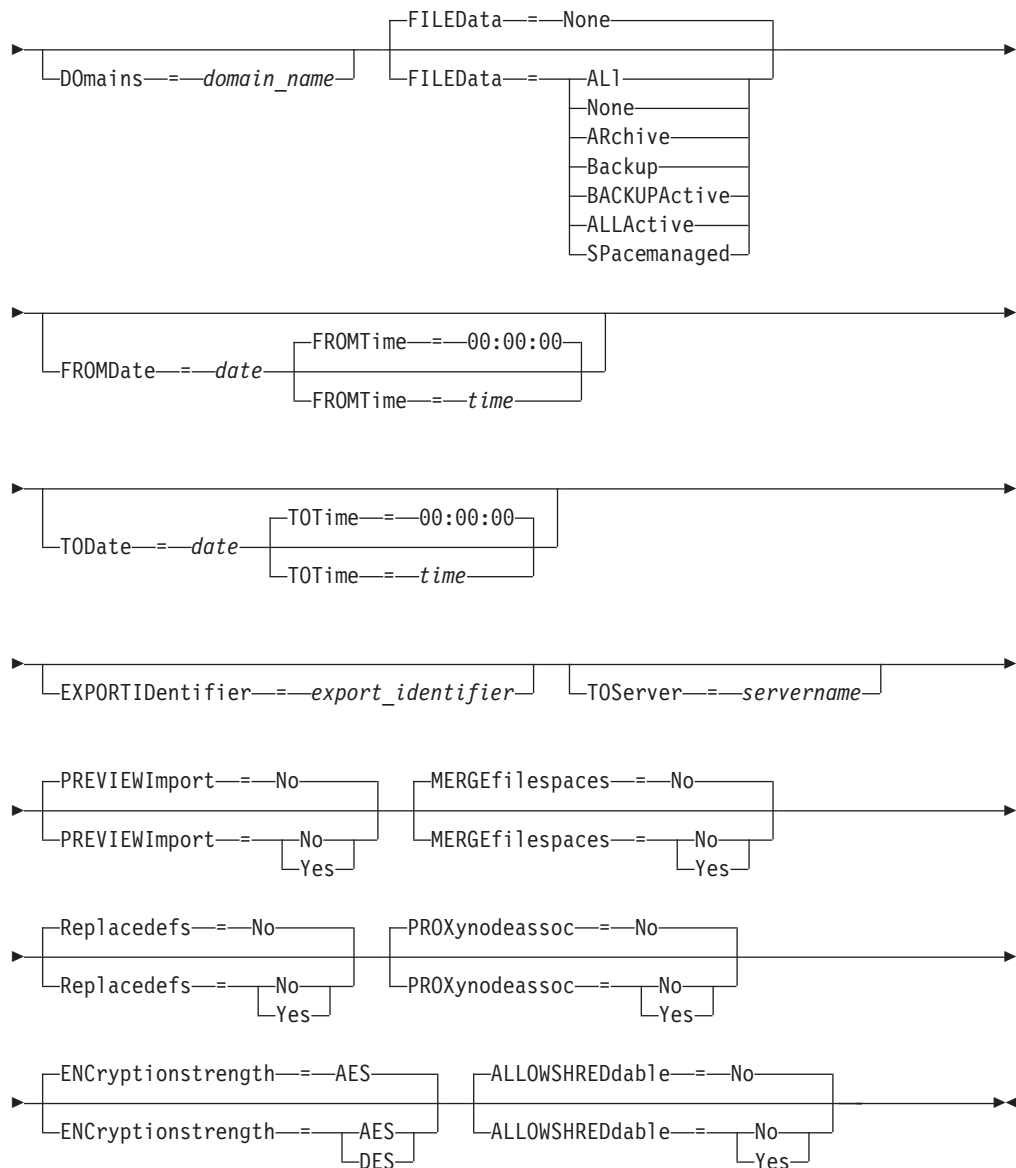
### Privilege class

To issue this command, you must have system privilege.

### Syntax



## EXPORT NODE— directly to another server



### Parameters

#### *node\_name*

Specifies the client node names for which information is to be exported. This parameter is optional. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. For each node entered, all file spaces in the file space, FSID, and Unicode enabled lists will be searched.

**Restriction:** If you specify a list of node names or node patterns, the server will not report the node names or node patterns that do not match any of the entries in the database. Check the summary statistics in the activity log to verify that the server exported all intended nodes.

#### **FILESpace**

Specifies the file spaces for which data is to be exported. This parameter is optional. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify a name.

**FSID**

Specifies the file spaces by using their file space IDs (FSIDs). The server uses the FSIDs to find the file spaces to export. To find the FSID for a file space, use the QUERY FILESPACE command. Separate multiple file space IDs with commas and no intervening spaces. This parameter is optional.

**UNIFILESpace**

Specifies the file spaces that are known to the server to be Unicode enabled. The server converts the names you enter from the server code page to the UTF-8 code page to find the file spaces to export. The success of the conversion depends on the actual characters in the name and the server's code page. Separate multiple names with commas and no intervening spaces. This parameter is optional.

**DOmains**

Specifies the policy domains from which nodes should be exported. This parameter is optional. Separate multiple names with commas and no intervening spaces. If you specify domains, Tivoli Storage Manager exports a node only if it belongs to one of the specified domains. You can use wildcard characters to specify a name.

**FILEData**

Specifies the type of files to export for all nodes. This parameter is optional. The default value is NONE.

If you are exporting to sequential media, 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, Tivoli Storage Manager requires two drives to export node information. The mount limit for the device class must be at least 2.

**Important:** If you export client nodes that are registered as TYPE=SERVER, specify ALL, ARCHIVE, or ALLACTIVE.

The following descriptions mention *active* and *inactive* backup file versions. An active backup file version is the most recent backup version for a file that still exists on the client workstation. All other backup file versions are called inactive copies. The values are:

**ALL**

The server exports all backup versions of files, all archived files, and all files migrated by a Tivoli Storage Manager for Space Management client.

**None**

The server does not export files, only node definitions.

**ARChive**

The server exports only archived files.

**Backup**

The server exports only backup versions, whether they are active or inactive.

**BACKUPActive**

The server exports only active backup versions. These active backup versions are the active versions in the Tivoli Storage Manager database at the time that the EXPORT command is issued.

**ALLActive**

The server exports all active backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space

## EXPORT NODE— directly to another server

Management client. The active backup versions are the active versions in the Tivoli Storage Manager database at the time that the EXPORT command is issued.

### SPacemanaged

The server exports only files that were migrated by a Tivoli Storage Manager for Space Management client.

### FROMDate

Specifies the earliest date for which objects to be exported were stored on the server. Objects that were stored on the server earlier than the specified date will not be exported. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

You can specify the date by using one of the values below:

| Value                             | Description  | Example         |
|-----------------------------------|--|-----------------|
| <i>MM/DD/YYYY</i>                 | A specific date  | 09/15/1998      |
| <b>TODAY</b>                      | The current date   | TODAY           |
| <b>TODAY-days</b> or <i>-days</i> | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

If this parameter is not specified, Tivoli Storage Manager exports all objects stored before the TODATE parameter and as qualified by the FILEDATA parameter. If no TODATE parameter is specified, then all data as qualified by the FILEDATA parameter is exported.

### TODate

Specifies the latest date for objects to be exported from the server. Objects stored on the server on a date later than the TODATE value are not exported. TODATE only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TODATE parameter when the FILEDATA parameter is set to NONE.

**Important:** If a TODATE parameter is specified without a TOTIME parameter, the server exports all objects inserted on or before the day specified by the TODATE parameter.

You can specify the date by using one of the values below:

| Value                             | Description  | Example         |
|-----------------------------------|--|-----------------|
| <i>MM/DD/YYYY</i>                 | A specific date  | 10/15/2006      |
| <b>TODAY</b>                      | The current date   | TODAY           |
| <b>TODAY-days</b> or <i>-days</i> | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

**Important:** If you specified the FROMDATE parameter, the value of TODATE must be later than or equal to the FROMDATE value. If the TODATE and FROMDATE are equal, then the TOTIME parameter must be later than the FROMTIME parameter.

**FROMTime**

Specifies the earliest time for which objects to be exported were stored on the server. When you specify FROMTIME, you must also use the FROMDATE parameter. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Objects that were stored on the server before the specified time and date will not be exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

You can specify the time by using one of the values below:

| Value                                | Description   | Example  |
|--------------------------------------|---|--|
| <i>HH:MM:SS</i>                      | A specific time   | 10:30:08   |
| <b>NOW</b>                           | The current time  | NOW  |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes specified. The FROMTIME+ can only be used with a FROMDATE before today. | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW+02:00 or FROMTIME=+02:00, the export operation only contains files that were put on the server after 7:00 on the FROMDATE that you specify. |
| <b>NOW-HH:MM</b><br><b>or -HH:MM</b> | The current time minus hours and minutes specified  | NOW-02:00 <b>or</b> -02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW-02:00 or FROMTIME=-2:00, the export includes files that were put on the server after 3:00.  |

**TOTime**

Specifies the latest time that objects to be exported were stored on the server. You must specify the TODATE parameter in order to use the TOTIME parameter. TOTIME only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TOTIME parameter if the FILEDATA parameter is set to NONE.

The default value for this parameter, when used with the TODATE parameter, is midnight minus one second (23:59:59).

**Important:** The value of the TOTIME and TODATE parameters must be later than the FROMDATE and the FROMTIME value.

You can specify the time by using one of the values below:

| Value                                | Description  | Example  |
|--------------------------------------|--|--|
| <i>HH:MM:SS</i>                      | A specific time                                    | 10:30:08   |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes specified. | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW+02:00, the export includes files that were stored from 01:00 until 07:00. |

## EXPORT NODE— directly to another server

| Value                 | Description   | Example   |
|-----------------------|---|---|
| NOW-HH:MM<br>or-HH:MM | The current time minus hours and minutes specified. | NOW-02:00 or -02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW-02:00, the export includes files that were stored from 01:00 until 03:00. |

### TOServer

Specifies the name of a server to which the export data will be sent directly over the network for immediate import.

**Important:** The target server must be defined on the originating server with the DEFINE SERVER command. The administrator that issues the export command must be defined with the same administrator name and password and have system authority on the target server.

When you specify TOSERVER, you cannot specify the DEVCLASS, VOLUMENAMES, and SCRATCH, USEDVOLUMELIST, and PREVIEW parameters.

### PREVIEWImport

Specifies whether to view how much data will be transferred, without actually moving any data. This information can be used to determine how much storage pool space is required on the target server. The default is NO.

Valid values are:

#### Yes

Specifies that you want to preview the results of the import operation on the target server, without importing the data. Information is reported to the server console and the activity log.

#### No

Specifies that you want the data to be imported on the target server without previewing the results.

### MERGEfilespace

Specifies whether Tivoli Storage Manager merges client files into existing file spaces on the target server (if they exist), or if Tivoli Storage Manager generates new file space names. The default is NO.

Valid values are:

#### Yes

Specifies that imported data on the target server is merged with the existing file space, if a file space with the same name already exists on the target server.

#### No

Specifies that Tivoli Storage Manager generates a new file space name for imported data on the target server if file spaces with the same name already exist.

### Replacedefs

Specifies whether to replace definitions (not file data) on the server. The default is NO.

Valid values are:



## EXPORT NODE— directly to another server

### Yes

Specifies that definitions are replaced on the server if definitions having the same name as those being imported already exist on the target server.

### No

Specifies that imported definitions are skipped if their names conflict with definitions that are already defined on the target server.

### PROXynodeassoc

Specifies if proxy node associations are exported. This parameter is optional. The default value is NO.

### ENCryptionstrength

Indicates which algorithm to use to encrypt passwords when exporting administrative and node records. This parameter is optional. The default value is AES. If you are exporting to a server that does not support AES, specify DES. Possible values are:

#### AES

Specifies the Advanced Encryption Standard.

#### DES

Specifies the Data Encryption Standard.

### ALLOWSHREDdable

Specifies whether data from a storage pool that enforces shredding is exported. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that the server does not export data from a storage pool that enforces shredding.

#### Yes

Specifies that the server does export from a storage pool that enforces shredding. The data on the export media will not be shredded.

**Restriction:** After an export operation finishes identifying files for export, any changes to the storage pool **ALLOWSHREDABLE** value is ignored. An export operation that is suspended retains the original **ALLOWSHREDABLE** value throughout the operation. You might want to consider cancelling your export operation if changes to the storage pool **ALLOWSHREDABLE** value jeopardize the operation. You can reissue the export command after any needed cleanup.

### EXPORTIDentifier

This optional parameter specifies the name that you select to identify this export operation. If you do not specify an identifier name, the server generates one for you. The export identifier name cannot be more than 64 characters, cannot contain wildcard characters, and is not case sensitive. You can use the identifier name to reference export operations in the QUERY EXPORT, SUSPEND EXPORT, RESTART EXPORT, or CANCEL EXPORT commands.

**Restriction:** You must specify the **TOSERVER** parameter if you are specifying the **EXPORTIDENTIFIER** parameter.

**EXPORTIDENTIFIER** is ignored if **FILEDATA=NONE**.

### Example: Export client node information and all client files

To export client node information and all client files for NODE1 directly to SERVERB, issue the following command:

## EXPORT NODE— directly to another server

```
export node node1 filedata=all toserver=serverb
```

### **Example: Export client node information and all client files for a specific date range**

To export client node information and all client files for NODE1 directly to SERVERB between February 1, 2009 and today.

```
export node node1 filedata=all toserver=serverb  
fromdate=02/01/2009 todate=today
```

### **Example: Export client node information and all client files for a specific date and time range**

To export client node information and all client files for NODE1 directly to SERVERB from 8:00 AM on February 1, 2009 until today at 8:00 AM, issue the following command:

```
export node node1 filedata=all toserver=serverb  
fromdate=02/01/2009 fromtime=08:00:00  
todate=today totime=08:00:00
```

### **Example: Export client node information and all client files for the past three days**

To export client node information and all client files for NODE1 directly to SERVERB for the past three days, issue the following command:

```
export node node1 filedata=all toserver=serverb  
fromdate=today -3
```

## EXPORT POLICY (Export policy information)

Use this command to export policy information from a Tivoli Storage Manager server to sequential media or directly to another server for immediate import.

### Important:

- If target and source server levels are not compatible, the operation may not work. See the *Administrator's Guide* for server compatibility requirements.
- Exporting data to a CENTERA device class is not supported. However, files stored in CENTERA storage pools can be exported.

The server exports 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 for each policy domain
- Client node associations, if the client node exists on the target server

You can use the QUERY ACTLOG command to view the status of the export operation. You can also view this information from the server console.

This command generates a background process that can be cancelled with the CANCEL PROCESS command. If you export policy information to sequential media and the background process is cancelled, the sequential media holding the exported data are incomplete and should not be used for importing data. If a server-to-server export background process is cancelled, a partial import may result. Evaluate any imported data on the target server to determine if you want to keep or delete the imported data. Review the import messages for details. To display information on background processes, use the QUERY PROCESS command.

The EXPORT POLICY command takes two forms: export directly to another server on the network, or export to sequential media. The syntax and parameters for each form are defined separately.

*Table 159. Commands related to EXPORT POLICY*

| Command        | Description  |
|----------------|--|
| CANCEL PROCESS | Cancels a background server process.                 |
| EXPORT ADMIN   | Copies administrative information to external media. |
| EXPORT NODE    | Copies client node information to external media.    |
| EXPORT SERVER  | Copies all or part of the server to external media.  |
| IMPORT POLICY  | Restores policy information from external media.     |
| QUERY ACTLOG   | Displays messages from the server activity log.      |
| QUERY PROCESS  | Displays information about background processes.     |

## EXPORT POLICY— to sequential media

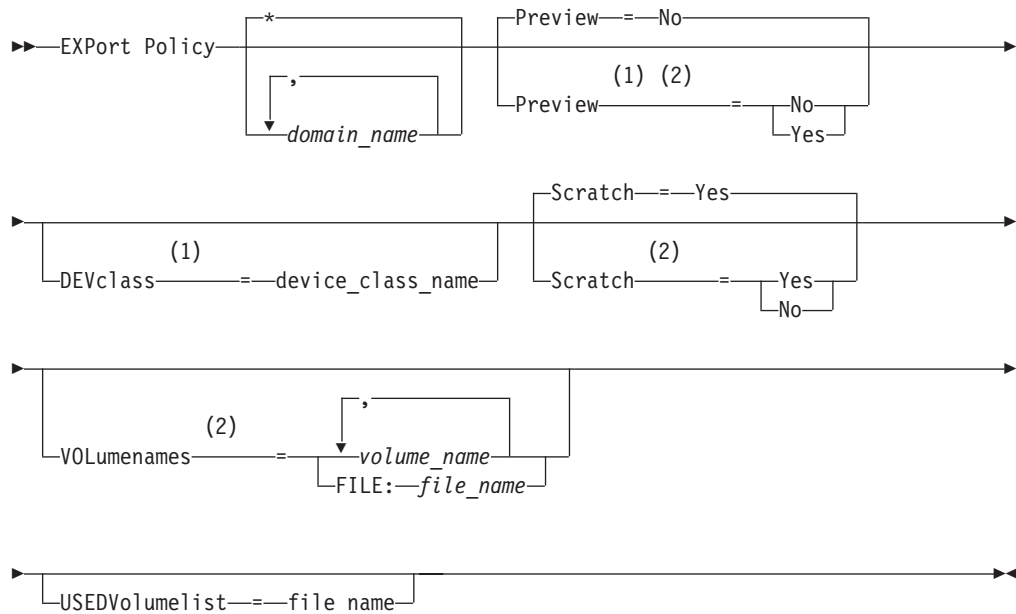
### EXPORT POLICY (Export policy information to sequential media)

Use this command to export policy information from an IBM Tivoli Storage Manager server to sequential media for later import to another server.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax



#### Notes:

- 1 If `PREVIEW=NO`, a device class must be specified.
- 2 If `PREVIEW=NO` and `SCRATCH=NO`, one or more volumes must be specified.

#### Parameters

##### *domain\_name*

Specifies the policy domains for which information is to be exported. This parameter is optional. The default is all policy domains. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names.

##### Preview

Specifies whether to preview the results of the export operation, without 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. This parameter is optional. The default value is `NO`. The values are:

##### No

Specifies that the policy information is to be exported. If you specify this value, you must also specify a device class.

##### Yes

Specifies that the operation will be previewed but not performed.

Information is reported to the server console and the activity log. If you specify this value, you do not need to specify a device class.

### DEVclass

Specifies the device class to which export data is to be written. This parameter is required if you specify PREVIEW=NO.

You cannot specify the DISK, NAS, or CENTERA device classes.

If all drives for the device class are busy when the export runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

**Tip:** You can export data to a storage pool on another server by specifying a device class whose device type is SERVER. For details about storing data on another server, see the *Administrator's Guide*.

### Scratch

Specifies whether scratch volumes can be used. The default value is YES.

Possible values are:

#### Yes

Specifies that scratch volumes can be used for export. If you also specify a list of volumes, scratch volumes are used only if there is not enough space on the volumes specified.

#### No

Specifies that scratch volumes cannot be used for export. To determine how many volumes you might need, you can run the command specifying PREVIEW=YES.

### VOLumenames

Specifies the volumes to be used to contain exported data. This parameter is optional, unless you specify SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, scratch volumes are used.

Possible values are:

*volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify  |
|-----------------|--|
| Tape            | 1–6 alphanumeric characters.   |
| FILE            | Any fully qualified file name string. For example:<br><br>/imdata/mt1. |
| OPTICAL         | 1–32 alphanumeric characters.  |
| REMOVABLEFILE   | 1–6 alphanumeric characters.   |
| SERVER          | 1–250 alphanumeric characters.   |

## EXPORT POLICY— to sequential media

### USEDVolumelist

Specifies the file where a list of volumes used in the export operation will be stored. This parameter is optional.

This file can be used as input to the import operation. This file will contain comment lines with the date and time the export was done, and the command issued to create the export.

**Attention:** If you specify an existing file, the file is overwritten.

### Example: Export policy information to specific tape volumes

From the server, export policy information to tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class.

```
export policy devclass=menu1  
volumenames=tape01,tape02,tape03
```

### Example: Export policy information to tape volumes listed in a file

From the server, export policy information to tape volumes that are listed in the following file:

TAPEVOL

This file contains the following lines:

```
TAPE01  
TAPE02  
TAPE03
```

Specify that these tape volumes be used by a device assigned to the MENU1 device class. Issue the following command:

```
export policy devclass=menu1 volumenames=file:tapevol
```

## EXPORT POLICY (Export a policy directly to another server)

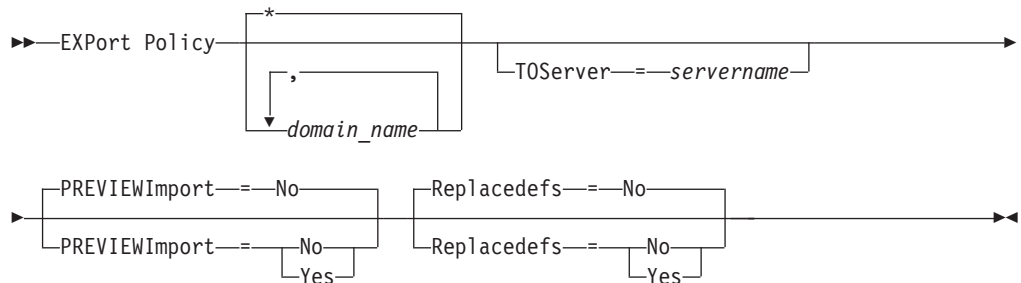
Use this command to export policy information directly to another server on the network. This results in an immediate import on the target server.

To monitor the progress of the import operation, you can issue a QUERY PROCESS command from the target server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *domain\_name*

Specifies the policy domains for which information is to be exported. This parameter is optional. The default is all policy domains. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names.

#### **TO Server**

Specifies the name of a server to which the export data will be sent directly over the network for immediate import.

**Important:** The target server must be defined on the originating server with the DEFINE SERVER command. The administrator that issues the export command must be defined with the same administrator name and password and have system authority on the target server.

When you specify TOSERVER, you cannot specify the DEVCLASS, VOLUMENAMES, and SCRATCH, USEDVOLUMELIST, and PREVIEW parameters.

#### **PREVIEWImport**

Specifies whether to view how much data will be transferred, without actually moving any data. This information can be used to determine how much storage pool space is required on the target server. The default is NO.

Valid values are:

##### **Yes**

Specifies that you want to preview the results of the import operation on the target server, without importing the data. Information is reported to the server console and the activity log.

## **EXPORT POLICY—directly to another server**

### **No**

Specifies that you want the data to be imported on the target server without previewing the results.

### **Replacedefs**

Specifies whether to replace definitions (not file data) on the server. The default is NO.

Valid values are:

### **Yes**

Specifies that definitions are replaced on the server if definitions having the same name as those being imported already exist on the target server.

### **No**

Specifies that imported definitions are skipped if their names conflict with definitions that are already defined on the target server.

## **Example: Export policy to another server**

To export policy information directly to SERVERB, issue the following command:

```
export policy replacedefs=yes toserver=othersrv
```



## EXPORT SERVER (Export server information)

Use this command to export all or part of the server control information and client file data (if specified) from the server to sequential media.

When you export server information to sequential media, you can later use the media to import the information to another server with a compatible device type.

### Important:

- If target and source server levels are not compatible, the operation might not work. For information about server compatibility requirements, see the *Administrator's Guide*.
- Exporting data to a CENTERA device class is not supported. However, files stored in CENTERA storage pools can be exported.

You also have the option of processing an export operation directly to another server on the network. This results in an immediate import process without the need for compatible sequential device types between the two servers.

You can export the following types of server information by issuing the EXPORT SERVER command:

- Policy domain definitions
- Policy set definitions
- Management class and copy group definitions
- Schedules defined for each policy domain
- Administrator definitions
- Client node definitions

You can optionally export the following types of data:

- File space definitions
- Access authorization information pertaining to the file spaces exported
- Backed-up, archived, and files that were migrated by a Tivoli Storage Manager for Space Management client

You cannot export nodes of type NAS. Export processing will exclude these nodes.

This command generates a background process that can be cancelled by the CANCEL PROCESS command. If you export server information to sequential media, and the background process is cancelled, the sequential media holding the exported data are incomplete and should not be used for importing data. If a server-to-server export background process is cancelled, a partial import may result. Evaluate any imported data on the target server to determine if you want to keep or delete the imported data. Review the import messages for details.

Issue the QUERY PROCESS command from the target server to monitor the progress of the import operation. Issue the QUERY EXPORT command to list all server-to-server export operations (that have a FILEDATA value other than NONE) that are running or suspended.

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.

The EXPORT SERVER command takes two forms: export directly to another server on the network, or export to sequential media. The syntax and parameters for each

## EXPORT SERVER

form are defined separately.

*Table 160. Commands related to EXPORT SERVER*

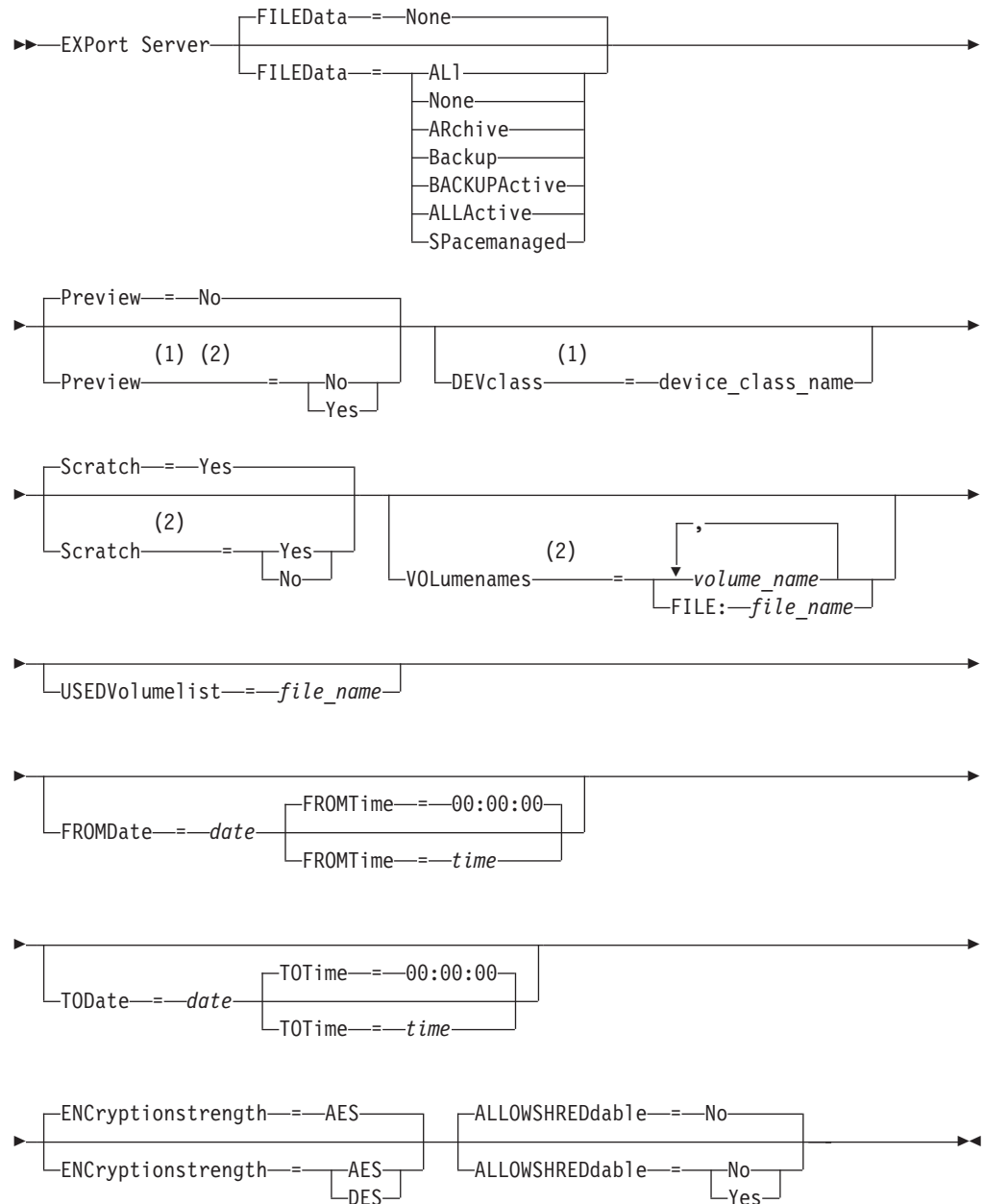
| Command           | Description   |
|-------------------|---|
| CANCEL EXPORT     | Deletes a suspended export operation                                    |
| CANCEL PROCESS    | Cancels a background server process.                                    |
| COPY ACTIVATEDATA | Copies active backup data.  |
| EXPORT ADMIN      | Copies administrative information to external media.                    |
| EXPORT NODE       | Copies client node information to external media.                       |
| EXPORT POLICY     | Copies policy information to external media.                            |
| IMPORT SERVER     | Restores all or part of the server from external media.                 |
| QUERY ACTLOG      | Displays messages from the server activity log.                         |
| QUERY EXPORT      | Displays the export operations that are currently running or suspended. |
| QUERY PROCESS     | Displays information about background processes.                        |
| RESTART EXPORT    | Restarts a suspended export operation.                                  |
| SUSPEND EXPORT    | Suspends a running export operation.                                    |

**EXPORT SERVER (Export a server to sequential media)**

You can export all or part of the server control information and client file data from a server to sequential media for later importing to another server.

**Privilege class**

To issue this command, you must have system privilege.

**Syntax****Notes:**

- 1 If PREVIEW=NO, a device class must be specified.
- 2 If PREVIEW=NO and SCRATCH=NO, one or more volumes must be specified.

### Parameters

#### FILEData

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.

If you are exporting to sequential media: the device class to access the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, two drives are needed to export server information. The mount limit for the device class must be set to at least 2.

The following descriptions mention *active* and *inactive* backup file versions. An active backup file version is the most recent backup version for a file that still exists on the client workstation. All other backup file versions are called inactive copies. The following values are available:

#### ALL

Tivoli Storage Manager exports all backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client.

#### None

Tivoli Storage Manager does not export files, only definitions.

#### ARchive

Tivoli Storage Manager exports only archived files.

#### Backup

Tivoli Storage Manager exports only backup versions, whether the versions are active or inactive.

#### BACKUPActive

Tivoli Storage Manager exports only active backup versions.

#### ALLActive

Tivoli Storage Manager exports all active backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client.

#### SPacemanaged

Tivoli Storage Manager exports only files that were migrated by a Tivoli Storage Manager for Space Management client.

#### Preview

Specifies whether you want to preview the results of the export operation, without exporting information. You can use this parameter to preview how many bytes of data are transferred, so that you can determine how many volumes are required. This parameter is optional. The default value is NO. The values are:

#### No

Specifies that the server information is to be exported. If you specify this value, you must also specify a device class.

#### Yes

Specifies that the operation will be previewed but not performed. Information is reported to the server console and the activity log. If you specify this value, you do not need to specify a device class.

#### DEVclass

Specifies the device class to which export data is to be written. This parameter is required if you specify PREVIEW=NO.

You cannot specify the DISK, NAS, or CENTERA device classes.

If all drives for the device class are busy when the export runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

**Tip:** You can export data to a storage pool on another server by specifying a device class whose device type is SERVER. For details about storing data on another server, see the *Administrator's Guide*.

### Scratch

Specifies whether scratch volumes can be used. The default value is YES. Possible values are:

#### Yes

Specifies that scratch volumes can be used for export. If you also specify a list of volumes, scratch volumes are used only if there is not enough space on the volumes specified.

#### No

Specifies that scratch volumes cannot be used for export. To determine how many volumes you might need, you can run the command specifying PREVIEW=YES.

### VOLumenames

Specifies the volumes to be used to contain exported data. This parameter is optional, unless you specify SCRATCH=NO and PREVIEW=NO. If you do not specify a volume name, scratch volumes are used.

Possible values are:

#### *volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

#### **FILE:***file\_name*

Specifies the name of a file that contains a list of volumes. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify  |
|-----------------|--|
| Tape            | 1–6 alphanumeric characters.   |
| FILE            | Any fully qualified file name string. For example:<br><br>/imdata/mt1. |
| OPTICAL         | 1–32 alphanumeric characters.  |
| REMOVABLEFILE   | 1–6 alphanumeric characters.   |
| SERVER          | 1–250 alphanumeric characters.   |

### USEDVolumelist

Specifies the file where a list of volumes used in the export operation will be stored. This parameter is optional.

This file can be used as input to the import operation. This file will contain comment lines with the date and time the export was done, and the command issued to create the export.

## EXPORT SERVER— to sequential media

**Attention:** If you specify an existing file, the file is overwritten.

### FROMDate

Specifies the earliest date for which objects to be exported were stored on the server. Objects that were stored on the server earlier than the specified date will not be exported. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

You can specify the date by using one of the values below:

| Value                  | Description  | Example         |
|------------------------|--|-----------------|
| MM/DD/YYYY             | A specific date  | 09/15/1998      |
| TODAY                  | The current date   | TODAY           |
| TODAY-days or<br>-days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

If this parameter is not specified, Tivoli Storage Manager exports all objects stored before the TODATE parameter and as qualified by the FILEDATA parameter. If no TODATE parameter is specified, then all data as qualified by the FILEDATA parameter is exported.

### TODate

Specifies the latest date for objects to be exported from the server. Objects stored on the server on a date later than the TODATE value are not exported. TODATE only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TODATE parameter when the FILEDATA parameter is set to NONE.

**Important:** If a TODATE parameter is specified without a TOTIME parameter, the server exports all objects inserted on or before the day specified by the TODATE parameter.

You can specify the date by using one of the values below:

| Value                  | Description  | Example         |
|------------------------|--|-----------------|
| MM/DD/YYYY             | A specific date  | 10/15/2006      |
| TODAY                  | The current date   | TODAY           |
| TODAY-days or<br>-days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

**Important:** If you specified the FROMDATE parameter, the value of TODATE must be later than or equal to the FROMDATE value. If the TODATE and FROMDATE are equal, then the TOTIME parameter must be later than the FROMTIME parameter.

### FROMTime

Specifies the earliest time for which objects to be exported were stored on the server. When you specify FROMTIME, you must also use the FROMDATE parameter. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Objects that were stored on the server before the specified time and date will

not be exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

You can specify the time by using one of the values below:

| Value                                | Description   | Example  |
|--------------------------------------|---|--|
| <i>HH:MM:SS</i>                      | A specific time   | 10:30:08   |
| <b>NOW</b>                           | The current time  | NOW  |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes specified. The FROMTIME+ can only be used with a FROMDATE before today. | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW+02:00 or FROMTIME=+02:00, the export operation only contains files that were put on the server after 7:00 on the FROMDATE that you specify. |
| <b>NOW-HH:MM</b><br><b>or -HH:MM</b> | The current time minus hours and minutes specified  | NOW-02:00 <b>or</b> -02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW-02:00 or FROMTIME=-2:00, the export includes files that were put on the server after 3:00.  |

### TOTime

Specifies the latest time that objects to be exported were stored on the server. You must specify the TODATE parameter in order to use the TOTIME parameter. TOTIME only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TOTIME parameter if the FILEDATA parameter is set to NONE.

The default value for this parameter, when used with the TODATE parameter, is midnight minus one second (23:59:59).

**Important:** The value of the TOTIME and TODATE parameters must be later than the FROMDATE and the FROMTIME value.

You can specify the time by using one of the values below:

| Value                               | Description   | Example  |
|-------------------------------------|---|--|
| <i>HH:MM:SS</i>                     | A specific time                                     | 10:30:08   |
| <b>NOW+HH:MM</b><br><b>or+HH:MM</b> | The current time plus hours and minutes specified.  | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW+02:00, the export includes files that were stored from 01:00 until 07:00. |
| <b>NOW-HH:MM</b><br><b>or-HH:MM</b> | The current time minus hours and minutes specified. | NOW-02:00 <b>or</b> -02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW-02:00, the export includes files that were stored from 01:00 until 03:00. |

## EXPORT SERVER— to sequential media

### ENCryptionstrength

Indicates which algorithm to use to encrypt passwords when exporting administrative and node records. This parameter is optional. The default value is AES. If you are exporting to a server that does not support AES, specify DES. Possible values are:

#### AES

Specifies the Advanced Encryption Standard.

#### DES

Specifies the Data Encryption Standard.

### ALLOWSHREDdable

Specifies whether data from a storage pool that enforces shredding is exported. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that data is not exported from a storage pool that enforces shredding.

#### Yes

Specifies that data can be exported from a storage pool that enforces shredding. The data on the export media will not be shredded.

### Example: Export a server to specific tape volumes

From the server, export server information to tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class.

```
export server devclass=menu1  
volumenames=tape01,tape02,tape03
```

### Example: Export a server to tape volumes listed in a file

From the server, export server information to tape volumes that are listed in the following file:

TAPEVOL

The file contains the following lines:

```
TAPE01  
TAPE02  
TAPE03
```

Specify that the tape volumes be used by a device assigned to the MENU1 device class. Issue the following command:

```
export server devclass=menu1 volumenames=file:tapevol
```



## EXPORT SERVER (Export server control information and client file data to another server)

Use this command to export all or part of the server control information and client file data directly to another server on the network. This results in an immediate import on the target server.

Server-to-server export operations that have a FILEDATA value other than NONE can be restarted after the operation is suspended. The server saves the state and status of the export operation so that it may be restarted from the point at which the operation failed or was suspended. The export operation can be restarted at a later date by issuing the RESTART EXPORT command. These export operations can be manually suspended as well as restarted. Therefore, if an export fails, it is automatically suspended if it has completed the transmitting definitions phase.

An export operation is suspended when any of the following conditions is detected:

- A SUSPEND EXPORT command is issued for the running export operation
- Segment preemption - the file being read for export is deleted by some other process
- Communication errors on a server-to-server export
- No available mount points
- Necessary volumes are unavailable
- I/O errors encountered

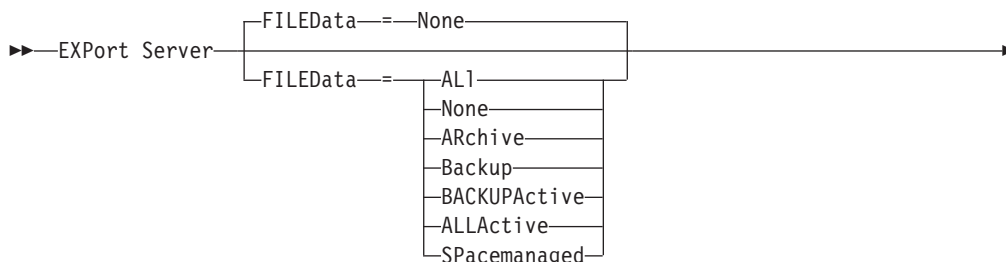
The export operation cannot be restarted if the export operation fails prior to transmitting the eligible node and filespace definitions to the target server. You must reenter the command to begin a new export operation.

Issue the QUERY PROCESS command from the target server to monitor the progress of the import operation. Issue the QUERY EXPORT command to list all server-to-server export operations (that have a FILEDATA value other than NONE) that are running or suspended.

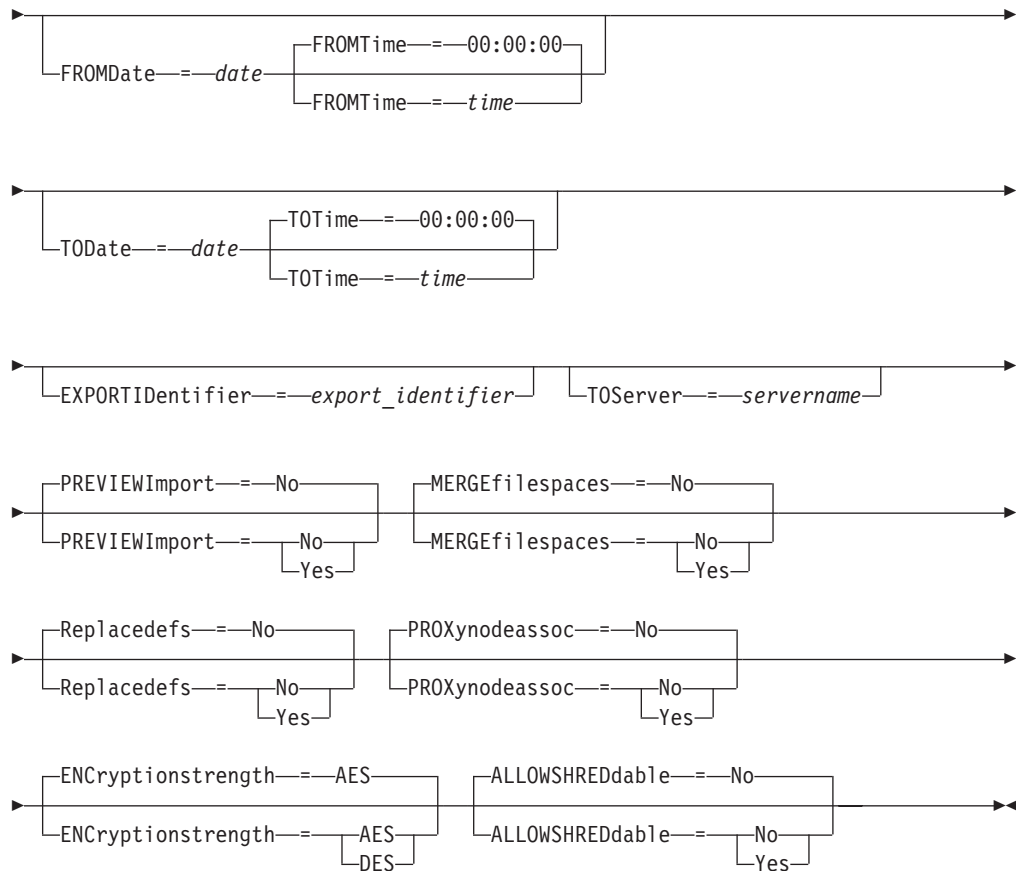
### Privilege class

To issue this command, you must have system privilege.

### Syntax



## EXPORT SERVER— directly to another server



### Parameters

#### FILEData

Specifies the type of files to export for all nodes defined to the server. This parameter is optional. The default value is NONE.

If you are exporting to sequential media: The device class to access the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, Tivoli Storage Manager requires two drives to export server information. You must set the mount limit for the device class to at least 2.

The following descriptions mention active and inactive backup file versions. An active backup file version is the most recent backup version for a file that still exists on the client workstation. All other backup file versions are called inactive copies. The values are:

#### ALL

Tivoli Storage Manager exports all backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client.

#### None

Tivoli Storage Manager does not export files, only definitions.

#### ARchive

Tivoli Storage Manager exports only archived files.

**Backup**

Tivoli Storage Manager exports only backup versions, whether they are active or inactive.

**BACKUPActive**

Tivoli Storage Manager exports only active backup versions.

**ALLActive**

Tivoli Storage Manager exports all active backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client.

**SPacemanaged**

Tivoli Storage Manager exports only files that were migrated by a Tivoli Storage Manager for Space Management client.

**FROMDate**

Specifies the earliest date for which objects to be exported were stored on the server. Objects that were stored on the server earlier than the specified date will not be exported. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

You can specify the date by using one of the values below:

| Value                                | Description  | Example         |
|--------------------------------------|--|-----------------|
| <i>MM/DD/YYYY</i>                    | A specific date  | 09/15/1998      |
| <b>TODAY</b>                         | The current date   | TODAY           |
| <b>TODAY-days</b> or<br><i>-days</i> | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

If this parameter is not specified, Tivoli Storage Manager exports all objects stored before the TODATE parameter and as qualified by the FILEDATA parameter. If no TODATE parameter is specified, then all data as qualified by the FILEDATA parameter is exported.

**TODate**

Specifies the latest date for objects to be exported from the server. Objects stored on the server on a date later than the TODATE value are not exported. TODATE only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TODATE parameter when the FILEDATA parameter is set to NONE.

**Important:** If a TODATE parameter is specified without a TOTIME parameter, the server exports all objects inserted on or before the day specified by the TODATE parameter.

You can specify the date by using one of the values below:

| Value             | Description      | Example    |
|-------------------|------------------|------------|
| <i>MM/DD/YYYY</i> | A specific date  | 10/15/2006 |
| <b>TODAY</b>      | The current date | TODAY      |

## EXPORT SERVER— directly to another server

| Value               | Description  | Example         |
|---------------------|--|-----------------|
| TODAY-days or -days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -3 or -3. |

**Important:** If you specified the FROMDATE parameter, the value of TODATE must be later than or equal to the FROMDATE value. If the TODATE and FROMDATE are equal, then the TOTIME parameter must be later than the FROMTIME parameter.

### FROMTime

Specifies the earliest time for which objects to be exported were stored on the server. When you specify FROMTIME, you must also use the FROMDATE parameter. This parameter only applies to client file data. This parameter has no effect on other information that may be exported, for example policy. Objects that were stored on the server before the specified time and date will not be exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

You can specify the time by using one of the values below:

| Value               | Description   | Example   |
|---------------------|---|---|
| HH:MM:SS            | A specific time   | 10:30:08  |
| NOW                 | The current time  | NOW   |
| NOW+HH:MM or +HH:MM | The current time plus hours and minutes specified. The FROMTIME+ can only be used with a FROMDATE before today. | NOW+02:00 or +02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW+02:00 or FROMTIME=+02:00, the export operation only contains files that were put on the server after 7:00 on the FROMDATE that you specify. |
| NOW-HH:MM or -HH:MM | The current time minus hours and minutes specified  | NOW-02:00 or -02:00.<br><br>If you issue this command at 5:00 with FROMTIME=NOW-02:00 or FROMTIME=-2:00, the export includes files that were put on the server after 3:00.  |

### TOTime

Specifies the latest time that objects to be exported were stored on the server. You must specify the TODATE parameter in order to use the TOTIME parameter. TOTIME only applies to client file data and does not affect other information that is being exported, such as policy. Tivoli Storage Manager ignores the TOTIME parameter if the FILEDATA parameter is set to NONE.

The default value for this parameter, when used with the TODATE parameter, is midnight minus one second (23:59:59).

**Important:** The value of the TOTIME and TODATE parameters must be later than the FROMDATE and the FROMTIME value.

## EXPORT SERVER— directly to another server

You can specify the time by using one of the values below:

| Value                               | Description   | Example  |
|-------------------------------------|---|--|
| <i>HH:MM:SS</i>                     | A specific time                                     | 10:30:08   |
| <i>NOW+HH:MM</i><br><i>or+HH:MM</i> | The current time plus hours and minutes specified.  | NOW+02:00 <b>or</b> +02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW+02:00, the export includes files that were stored from 01:00 until 07:00. |
| <i>NOW-HH:MM</i><br><i>or-HH:MM</i> | The current time minus hours and minutes specified. | NOW-02:00 <b>or</b> -02:00.<br><br>If you issue this command at 05:00 with FROMTIME=01:00 and TOTIME=NOW-02:00, the export includes files that were stored from 01:00 until 03:00. |

### TOServer

Specifies the name of a server to which the export data will be sent directly over the network for immediate import.

**Important:** The target server must be defined on the originating server with the DEFINE SERVER command. The administrator that issues the export command must be defined with the same administrator name and password and have system authority on the target server.

When you specify TOSERVER, you cannot specify the DEVCLASS, VOLUMENAMES, and SCRATCH, USEDVOLUMELIST, and PREVIEW parameters.

### PREVIEWImport

Specifies whether to view how much data will be transferred, without actually moving any data. This information can be used to determine how much storage pool space is required on the target server. The default is NO.

Valid values are:

#### Yes

Specifies that you want to preview the results of the import operation on the target server, without importing the data. Information is reported to the server console and the activity log.

#### No

Specifies that you want the data to be imported on the target server without previewing the results.

### MERGEfilespace

Specifies whether Tivoli Storage Manager merges client files into existing file spaces on the target server (if they exist), or if Tivoli Storage Manager generates new file space names. The default is NO.

Valid values are:

#### Yes

Specifies that imported data on the target server is merged with the existing file space, if a file space with the same name already exists on the target server.

## EXPORT SERVER— directly to another server

### No

Specifies that Tivoli Storage Manager generates a new file space name for imported data on the target server if file spaces with the same name already exist.

### Replacedefs

Specifies whether to replace definitions (not file data) on the server. The default is NO.

Valid values are:

### Yes

Specifies that definitions are replaced on the server if definitions having the same name as those being imported already exist on the target server.

### No

Specifies that imported definitions are skipped if their names conflict with definitions that are already defined on the target server.

### PROXynodeassoc

Specifies if proxy node associations are exported. This parameter is optional. The default value is NO.

### ENCryptionstrength

Indicates which algorithm to use to encrypt passwords when exporting administrative and node records. This parameter is optional. The default value is AES. If you are exporting to a server that does not support AES, specify DES. Possible values are:

### AES

Specifies the Advanced Encryption Standard.

### DES

Specifies the Data Encryption Standard.

### ALLOWSHREDdable

Specifies whether data from a storage pool that enforces shredding is exported. This parameter is optional. The default value is NO. Possible values are:

### No

Specifies that the server does not allow data to be exported from a storage pool that enforces shredding.

### Yes

Specifies that the server allows data to be exported from a storage pool that enforces shredding. The data on the export media will not be shredded.

**Important:** After an export operation finishes identifying files for export, any changes to the storage pool ALLOWSHREDABLE value is ignored. An export operation that is suspended retains the original ALLOWSHREDABLE value throughout the operation. You might want to consider cancelling your export operation if changes to the storage pool ALLOWSHREDABLE value jeopardize the operation. You can reissue the export command after any needed cleanup.

### EXPORTIDentifier

This optional parameter specifies the name that you selected to identify this export operation. If you do not specify a command name, the server generates one for you. The export identifier name cannot be more than 64 characters, cannot contain wildcard characters, and is not case sensitive. You can use the identifier name to reference export operations in the QUERY EXPORT,

## EXPORT SERVER— directly to another server

SUSPEND EXPORT, RESTART EXPORT, or CANCEL EXPORT commands.  
EXPORTIDENTIFIER is ignored if FILEDATA=NONE or if  
PREVIEWIMPORT=YES.

If you are specifying the EXPORTIDENTIFIER parameter, you must specify the  
TOSERVER parameter.

### **Example: Export server information directly to another server**

To export server information directly to SERVERB, issue the following command.

```
export server filedata=all toserver=serverb
```

### **Example: Export server information directly to another server using a date range**

To export directly to SERVERB between February 1, 2009 and today, issue the following command.

```
export server filedata=all toserver=serverb  
fromdate=02/01/2009 todate=today
```

### **Example: Export server information and client file data directly to another server using a date and time range**

To export directly to SERVERB from 8:00 a.m. on February 1, 2009 until today at 8:00 a.m., issue the following command.

```
export server filedata=all toserver=serverb  
fromdate=02/01/2009 fromtime=08:00:00  
todate=today totime=08:00:00
```

## EXTEND DBSPACE (Increase space for the database)

Use this command to increase space for the database by adding directories for the database to use.

### Privilege class

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

### Syntax

```

▶▶ EXTEND DBSPACE db_directory

```

### Parameters

#### *db\_directory* (Required)

Specifies the directories for database storage. The directories must be empty and accessible by the user ID of the database manager. A directory name must be a fully qualified name. The maximum length of the name is 175 characters. Enclose the name in quotation marks if it contains embedded blanks, an equal sign, or other special characters.

### Example: Add directories to the storage space for the database

Add two directories (/tsm\_db/stg1 and tsm\_db/stg2) under the /tsm\_db directory to the storage space for the database. Issue the command:

```
extend dbspace /tsm_db/stg1,/tsm_db/stg2
```

### Related commands

Table 161. Commands related to EXTEND DBSPACE

| Command       | Description  |
|---------------|--|
| QUERY DB      | Displays allocation information about the database.                    |
| QUERY DBSPACE | Displays information about the storage space defined for the database. |



---

## GENERATE commands

Use the GENERATE BACKUPSET command to generate a backup set for a selected filesystem or client node.

- “GENERATE BACKUPSET (Generate a backup set of a client’s data)” on page 476
- “GENERATE BACKUPSETTOC (Generate a table of contents for a backup set)” on page 484

## GENERATE BACKUPSET (Generate a backup set of a client's data)

Use this command to generate a backup set for a backup-archive client node. A *backup set* is a collection of a client's active backed up data, stored and managed as a single object, on specific media, in server storage. Although you can create a backup set for any client node, a backup set can only be *used* by a backup-archive client.

**Restriction:** You cannot generate a backup set for a NAS node.

The server creates copies of active versions of a client's backed up objects that are within the one or more file spaces specified with this command, and consolidates them onto sequential media. Currently, the backup object types supported for backup sets include directories and files only.

The backup-archive client node can restore its backup set from the server as well as from the media to which the backup set was written.

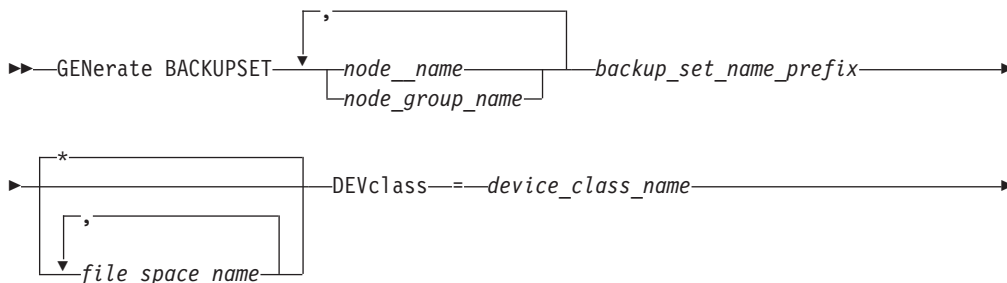
This command generates a background process that can be cancelled with the CANCEL PROCESS command. If the background process created by this command is cancelled, the media may not contain a complete backup set. You can use the QUERY PROCESS command to display information about the background process that is created by this command.

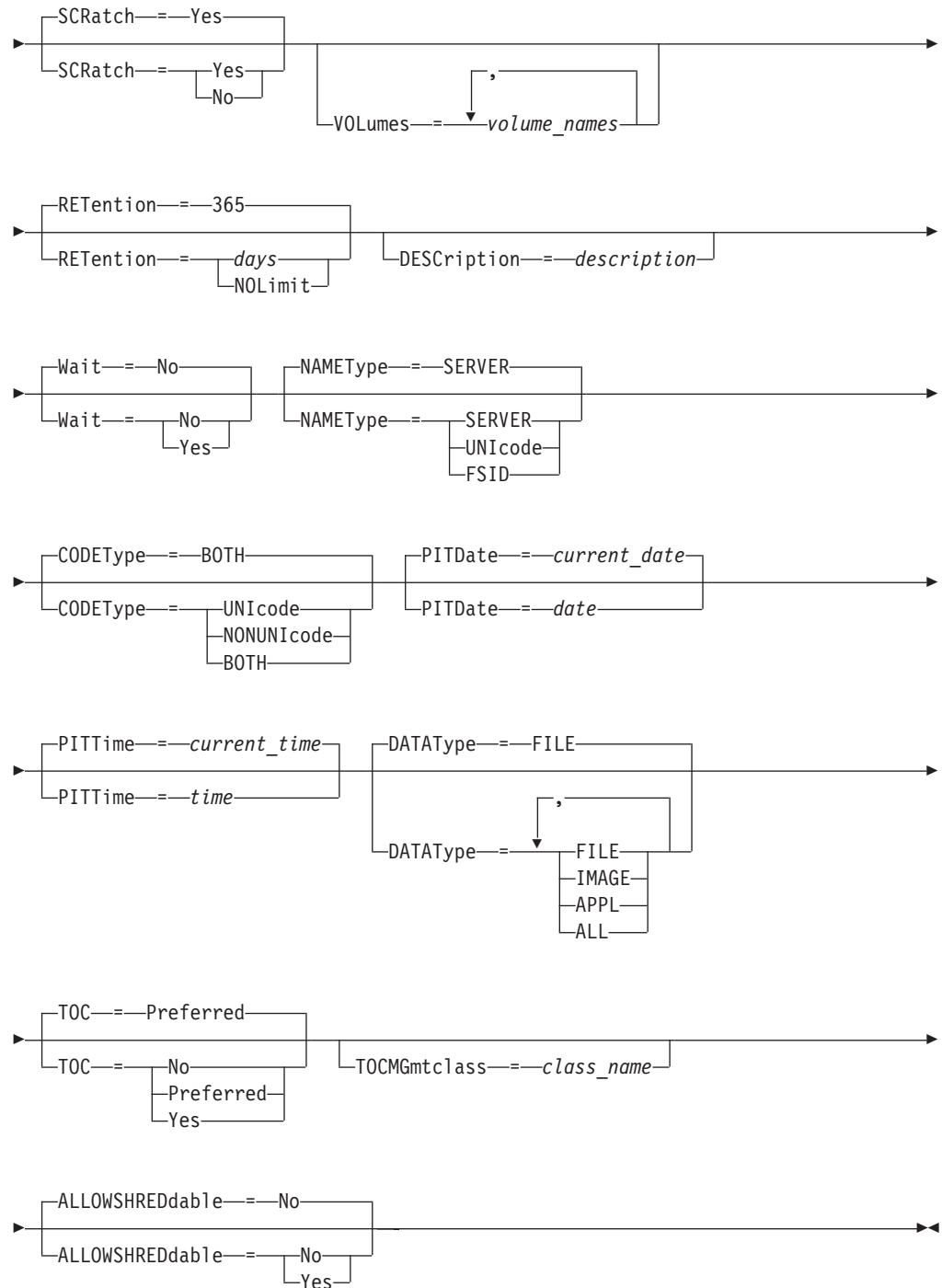
**Tip:** When Tivoli Storage Manager generates a backup set, you can improve performance if the primary storage pools containing the client data are collocated. If a primary storage pool is collocated, client node data is likely to be on fewer tape volumes than it would be if the storage pool were not collocated. With collocation, less time is spent searching database entries, and fewer mount operations are required.

### Privilege class

To issue this command, you must have system privilege or policy privilege for the domain to which the client node is assigned.

### Syntax





## Parameters

### *node\_name* or *node\_group\_name* (Required)

Specifies the name of the client node and node groups whose data is contained in the backup set. To specify multiple node names and node group names, separate the names with commas and no intervening spaces. You can use wildcard characters with node names but not with node group names. When

## GENERATE BACKUPSET

multiple node names are specified, the server generates a backup set for each node and places all of the backup sets together on a single set of output volumes.

### *backup\_set\_name\_prefix* (Required)

Specifies the name of the backup set for the client node. The maximum length of the name is 30 characters.

When you select a name, Tivoli Storage Manager adds a suffix to construct your backup set name. For example, if you name your backup set *mybackupset*, Tivoli Storage Manager adds a unique number such as 3099 to the name. The backup set name is then identified to Tivoli Storage Manager as *mybackupset.3099*. To later display information about this backup set, you can include a wildcard with the name, such as *mybackupset.\** or you can specify the fully qualified name, such as *mybackupset.3099*.

When multiple node names or node group names are specified, the server generates a backup set for each node or node group and places all the backup sets on a single set of output volumes. Each backup set is given the same fully qualified name consisting of the *backup\_set\_name\_prefix* and a suffix determined by the server.

### *file\_space\_name*

Specifies the names of one or more file spaces that contain the data to be included in the backup set. This parameter is optional. The file space name you specify can contain wildcard characters. You can specify more than one file space by separating the names with commas and no intervening spaces. If you do not specify a file space, data from all the client node's backed-up and active file spaces is included in the backup set.

For a server that has clients with support for Unicode-enabled file spaces, you can enter *either* a file space name or a file space ID (FSID). If you enter a file space name, you may need to have the server convert the file space name that you enter. For example, you may need to have the server convert the name you enter from the server's code page to Unicode. See the NAMETYPE parameter for details. If you do not specify a file space name, or specify only a single wildcard character for the name, you can use the CODETYPE parameter to limit the operation to Unicode file spaces or to non-Unicode file spaces.

### DEVclass (Required)

Specifies the name of the device class for the volumes to which the backup set is written. The maximum length of the name is 30 characters.

**Restriction:** You cannot specify a device class with a device type of NAS or CENTERA.

### SCRatch

Specifies whether to use scratch volumes for the backup set. If you include a list of volumes using the VOLUMES parameter, the server uses scratch volumes only if the data cannot be contained in the volumes you specify. The default is SCRATCH=YES. The values are:

#### YES

Specifies to use scratch volumes for the backup set.

#### NO

Specifies not to use scratch volumes for the backup set.

### VOLumes

Specifies the names of one or more volumes that will contain the backup set.

This parameter is optional. You can specify more than one volume by separating each volume with a comma, with no intervening spaces.

If you do not specify this parameter, scratch volumes are used for the backup set.

**RETention**

Specifies the number of days to retain the backup set on the server. You can specify an integer from 0 to 30000. The default is 365 days. The values are:

*days*

Specifies the number of days to retain the backup set on the server.

**NOLimit**

Specifies that the backup set should be retained on the server indefinitely.

If you specify **NOLIMIT**, the server retains the volumes containing the backup set forever, unless a user or administrator deletes the volumes from server storage.

**DESCription**

Specifies the description to associate with the backup set. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters.

**Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is NO. The values are:

**Yes**

Specifies the command processes in the foreground. Messages created are not displayed until the command completes processing. You cannot specify WAIT=YES from the server console.

**No**

Specifies that the command processes in the background. Use the QUERY PROCESS command to monitor the background processing of this command.

**NAMEType**

Specify how you want the server to interpret the file space names that you enter. This parameter is useful when the server has clients with support for Unicode-enabled file spaces. You can use this parameter for Tivoli Storage Manager clients using Windows, NetWare, or Macintosh OS X operating systems.

Use this parameter only when you enter a partly or fully qualified file space name. The default value is SERVER. Possible values are:

**SERVER**

The server uses the server's code page to interpret the file space names.

**UNICODE**

The server converts the file space name entered from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the name and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

**FSID**

The server interprets the file space names as their file space IDs (FSIDs).

## GENERATE BACKUPSET

**Important:** Use care when specifying this parameter if multiple node names are also specified. Different nodes may use the same file space ID for different file spaces, or different file space IDs for the same file space name. Therefore, specifying a file space ID as the file space names can result in the wrong data being written to the backup set for some nodes.

### CODEType

Specify what type of file spaces are to be included in the operation. The default is BOTH, meaning that file spaces are included regardless of code page type. Use this parameter only when you enter a single wildcard character for the file space name or when you do not specify any file space names. Possible values are:

#### UNICODE

Include only file spaces that are in Unicode.

#### NONUNICODE

Include only file spaces that are not in Unicode.

#### BOTH

Include file spaces regardless of code page type.

### PITDate

Specifies that files that were active on the specified date and that are still stored on the Tivoli Storage Manager server are to be included in the backup set, even if they are inactive at the time you issue the command. This parameter is optional. The default is the date on which the GENERATE BACKUPSET command is run. You can specify the date using one of the following values:

| Value               | Description                           | Example   |
|---------------------|---------------------------------------|---|
| MM/DD/YYYY          | A specific date                       | 09/15/1998  |
| TODAY               | The current date                      | TODAY   |
| TODAY-days or -days | The current date minus days specified | TODAY-7 or -7.<br><br>To include files that were active a week ago, specify PITDATE=TODAY-7 or PITDATE=-7 |
| EOLM                | The last day of the previous month    | EOLM  |
| BOTM                | The first day of the current month    | BOTM  |

### PITTime

Specifies that files that were active on the specified time and that are still stored on the Tivoli Storage Manager server are to be included in the backup set, even if they are inactive at the time you issue the command. This parameter is optional. If a PITDate was specified, the default is midnight (00:00:00); otherwise the default is the time at which the GENERATE BACKUPSET command is started. You can specify the time using one of the following values:

| Value    | Description                               | Example  |
|----------|---|----------|
| HH:MM:SS | A specific time on the specified PIT date | 12:33:28 |

| Value               | Description   | Example   |
|---------------------|---|---|
| NOW                 | The current date on the specified PIT date                        | NOW   |
| NOW+HH:MM or +HH:MM | The current time plus hours and minutes on the specified PIT date | NOW+03:00 or +03:00<br><br>If you issue this command at 9:00 with PITTIME=NOW+03:00 or PITTIME=+03:00. Tivoli Storage Manager includes files that were active at 12:00 on the PIT date. |

## DATATYPE

Specifies that backup sets containing the specified types of data that are to be generated. This parameter is optional. The default is that file level backup sets are to be generated. To specify multiple data types, separate data types with commas and no intervening spaces.

The server generates a backup set for each data type and places all the backup sets on a single set of output volumes. Each backup set is given the same fully qualified name consisting of the *backup\_set\_name\_prefix* and a suffix determined by the server. However, each backup set will have a different data type, as shown by the QUERY BACKUPSET command. Possible values are:

### ALL

Specifies that backup sets for all types of data (file level, image, and application) that have been backed up on the server are to be generated.

### FILE

Specifies that a file level backup set is to be generated. File level backup sets contain files and directories backed up by the backup client. If no files or directories have been backed up by the backup client, a file level backup set is not generated. This is the default.

### IMAGE

Specifies that an image backup set is to be generated. Image backup sets contain images created by the backup client BACKUP IMAGE command. Image backup sets are generated only if an image has been backed up by the backup client.

### APPL

Specifies that an application backup set is to be generated. Application backup sets contain data backed up to the Tivoli Storage Manager server by applications using the Tivoli Storage Manager application programming interface. This parameter applies only if the Tivoli Storage Manager server was upgraded from Tivoli Storage Manager Express and only for the Tivoli Storage Manager Express versions of Data Protection for Exchange and Data Protection for SQL. Application backup sets are generated only if application data has been backed up to the server.

## TOC

Specifies whether a table of contents (TOC) is saved for each file level backup set. Tables of contents are always saved for backup sets containing image or application data. The TOC parameter is ignored when generating image and application backup sets. A table of contents will always be generated for image and application backup sets.

Consider the following in determining whether you want to save a table of contents:

## GENERATE BACKUPSET

- If a table of contents is saved for a backup set, you can use the Tivoli Storage Manager Web backup-archive client to examine the entire file system tree and choose files and directories to restore. To create a table of contents, you must define the TOCDESTINATION attribute in the backup copy group for the management class that is specified by the **TOCMGMTCLASS** parameter. Creating a table of contents requires additional processing, storage pool space, and possibly a mount point during the backup set operation.
- If a table of contents is not saved for a backup set, you can still restore individual files or directory trees using the backup-archive client RESTORE BACKUPSET command, if you know the fully qualified name of each file or directory to be restored.

To display the contents of backup sets, you can also use the QUERY BACKUPSETCONTENTS command.

This parameter is optional. Possible values are:

### No

Specifies that table of contents information is not saved for file level backup sets.

### Preferred

Specifies that table of contents information should be saved for file level backup sets. This is the default. However, a backup set does not fail just because an error occurs during creation of the table of contents.

### Yes

Specifies that table of contents information must be saved for each file level backup set. A backup set fails if an error occurs during creation of the table of contents.

### TOCMGmtclass

Specifies the name of the management class to which the table of contents should be bound. If you do not specify a management class, the table of contents is bound to the default management class for the policy domain to which the node is assigned. In this case, creation of a table of contents requires that you define the TOCDESTINATION attribute in the backup copy group for the specified management class.

### ALLOWSHREDdable

Specifies whether data from a storage pool that enforces shredding is included in the backup set. This parameter is optional. Possible values are:

### No

Specifies that data from a storage pool that enforces shredding is not included in the backup set. This is the default.

### Yes

Specifies that data from a storage pool that enforces shredding can be included in the backup set. The data on the backup set media will not be shredded.

## Example: Generate a backup set for a file space

Generate a backup set of a file space called /srvr that belongs to client node JANE. Name the backup set PERS\_DATA and retain it for 75 days. Specify that volumes VOL1 and VOL2 contain the data for the backup set. The volumes are to be read by a device that is assigned to the AGADM device class. Include a description.



```
generate backupset jane pers_data /srvr devclass=agadm
retention=75 volumes=vol1,vol2
description="area 51 base image"
```

### Example: Generate a backup set of a Unicode-enabled file space

Generate a backup set of the Unicode-enabled file space, \\joe\c\$, that belongs to client node JOE. Name the backup set JOES\_DATA. Specify that volume VOL1 contain the data for the backup set. The volume is to be read by a device that is assigned to the AGADM device class. Have the server convert the \\joe\c\$ file space name from the server code page to the UTF-8 code page.

```
generate backupset joe joes_data \\joe\c$ devclass=agadm
volumes=vol1 nametype=unicode
```

### Related commands

Table 162. Commands related to GENERATE BACKUPSET

| Command                 | Description  |
|-------------------------|--|
| COPY ACTIVATEDATA       | Copies active backup data.   |
| DEFINE COPYGROUP        | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE NODEGROUP        | Defines a group of nodes.  |
| DEFINE NODEGROUPMEMBER  | Adds a client node to a node group.  |
| DEFINE BACKUPSET        | Defines a previously generated backup set to a server.                                     |
| DELETE BACKUPSET        | Deletes a backup set.  |
| DELETE NODEGROUP        | Deletes a node group.  |
| DELETE NODEGROUPMEMBER  | Deletes a client node from a node group.   |
| QUERY BACKUPSET         | Displays backup sets.  |
| GENERATE BACKUPSETTOC   | Generates a table of contents for a backup set.  |
| QUERY NODEGROUP         | Displays information about node groups.  |
| QUERY BACKUPSETCONTENTS | Displays contents contained in backup sets.  |
| UPDATE BACKUPSET        | Updates a retention value associated with a backup set.                                    |
| UPDATE COPYGROUP        | Changes one or more attributes of a copy group.  |
| UPDATE NODEGROUP        | Updates the description of a node group.   |

## GENERATE BACKUPSETTOC (Generate a table of contents for a backup set)

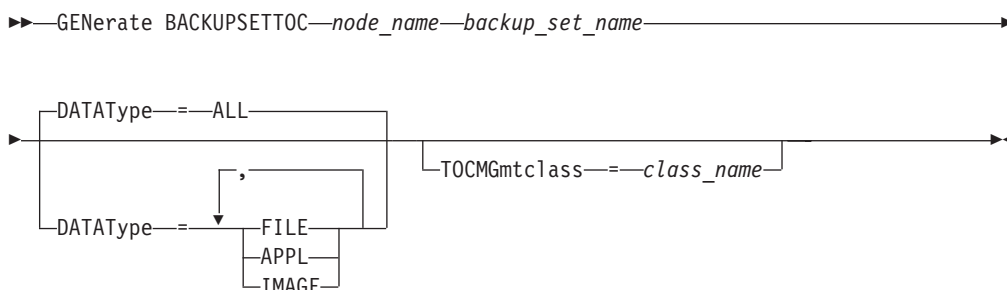
Use this command to generate a table of contents for a backup set that does not already have one. The backup-archive client uses the table of contents to display the backup set, which allows users to select individual files to be restored from the backup set.

Creating a table of contents for a backup set requires storage pool space and possibly one or more mount points during the creation operation.

### Privilege class

To issue this command, you must have system privilege or policy privilege for the domain to which the client node is assigned.

### Syntax



### Parameters

#### *node\_name* (Required)

Specifies the name of the client node whose data is contained in the backup set. You cannot use wildcard characters to specify a name, nor can you specify a list of client node names.

#### *backup\_set\_name* (Required)

Specifies the name of the backup set for the client node. You cannot use wildcard characters to specify a name, nor can you specify a list of backup set names.

#### **DATATYPE**

Specifies the type of data to be included in the table of contents. This parameter is optional. By default, all data is included. To specify multiple data types, separate the data types with commas and no intervening spaces. Possible values are:

##### **ALL**

Specifies that the table of contents includes all types of data (file-level, image, and application) stored in the backup set. This is the default.

##### **FILE**

Specifies that the table of contents includes only file-level data. File-level data consists of files and directories backed up by the backup-archive client. If the backup set contains no files or directories, the table of contents is not generated.

##### **APPL**

Specifies that the table of contents will include only application data.

Application data consists of data backed up to the Tivoli Storage Manager server by applications using the Tivoli Storage Manager application programming interface. This parameter applies only if the Tivoli Storage Manager server was upgraded from Tivoli Storage Manager Express and only for the Tivoli Storage Manager Express versions of Data Protection for Exchange and Data Protection for SQL. If the backup set contains no application data, the table of contents is not generated.

### IMAGE

Specifies that the table of contents will include only image backups. Image backups consist of file system images created by the backup client BACKUP IMAGE command. If the backup set contains no image backups, the table of contents will not be generated.

### TOCMgmtclass

Specifies the name of the management class to which the table of contents should be bound. If you do not specify a management class, the table of contents is bound to the default management class for the policy domain to which the node is assigned. If you create a table of contents you must define the TOCDESTINATION attribute in the backup copy group for the specified management class.

## Example: Generate a table of contents

Generate a table of contents for a backup set named PROJX\_DATA that contains the data for client node GARY. The table of contents is to be bound to the default management class.

```
generate backupsettoc gary projx_data
```

## Related commands

Table 163. Commands related to GENERATE BACKUPSETTOC

| Command                 | Description  |
|-------------------------|--|
| COPY ACTIVATEDATA       | Copies active backup data.   |
| DEFINE COPYGROUP        | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE NODEGROUP        | Defines a group of nodes.  |
| DEFINE NODEGROUPMEMBER  | Adds a client node to a node group.  |
| DEFINE BACKUPSET        | Defines a previously generated backup set to a server.                                     |
| DELETE BACKUPSET        | Deletes a backup set.  |
| DELETE NODEGROUP        | Deletes a node group.  |
| DELETE NODEGROUPMEMBER  | Deletes a client node from a node group.   |
| GENERATE BACKUPSET      | Generates a backup set of a client's data.   |
| QUERY BACKUPSET         | Displays backup sets.  |
| QUERY NODEGROUP         | Displays information about node groups.  |
| QUERY BACKUPSETCONTENTS | Displays contents contained in backup sets.  |
| UPDATE BACKUPSET        | Updates a retention value associated with a backup set.                                    |
| UPDATE COPYGROUP        | Changes one or more attributes of a copy group.  |

## GENERATE BACKUPSETTOC

*Table 163. Commands related to GENERATE BACKUPSETTOC (continued)*

| Command          | Description                              |
|------------------|--|
| UPDATE NODEGROUP | Updates the description of a node group. |

---

## GRANT commands

Use the GRANT command to grant appropriate administrator privileges or access.

- “GRANT AUTHORITY (Add administrator authority)” on page 488
- “GRANT PROXYNODE (Grant proxy authority to a client node)” on page 492

## GRANT AUTHORITY (Add administrator authority)

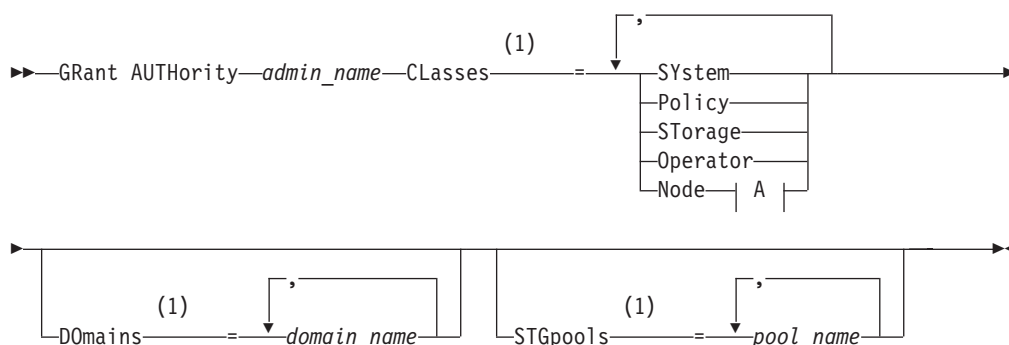
Use this command to grant an administrator one or more administrative privilege classes, and authority to access client nodes.

You cannot grant restricted privilege to an unrestricted policy or unrestricted storage administrator. 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.

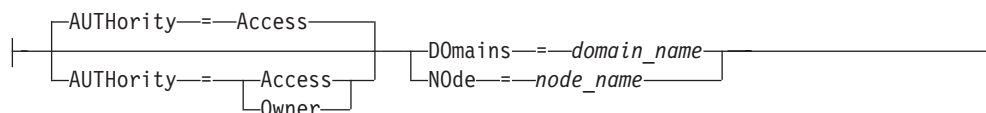
### Privilege class

To issue this command, you must have system privilege.

### Syntax



#### A:



#### Notes:

- 1 You must specify one or more of these parameters.

### Parameters

#### *admin\_name* (Required)

Specifies the name of the administrator being granted an administrative privilege class.

#### CLASSES

Specifies one or more privilege classes to grant to an administrator. This parameter is required, except when you specify the STGPools parameter. You can specify more than one privilege class by separating each with a comma. Possible classes are:

#### System

Specifies that you want to grant system privilege to an administrator. A system administrator has the highest level of authority in Tivoli Storage Manager. A system administrator can issue any administrative command

and has authority to manage all policy domains and all storage pools. Do not specify additional privilege classes or the DOMAINS or STGPools parameters when granting system privilege to an administrator. Only a system administrator can grant authority to other administrators.

**Policy**

Specifies that you want to grant policy privilege to an administrator. If you do not specify the DOMAINS parameter, 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. Use the GRANT AUTHORITY command with CLASSES=POLICY and no DOMAINS parameter to upgrade a restricted policy administrator to an unrestricted policy administrator.

**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. Using the GRANT AUTHORITY command with CLASSES=STORAGE and no STGPools parameter upgrades a restricted storage administrator to an unrestricted storage administrator.

**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.

**Node**

Specifies that you want to grant a node privilege to a user. A user with client node privilege can remotely access a web backup-archive client with an administrative user ID and password if they have been given owner authority or access authority. Access authority is the default for a node privilege class.

**Attention:** When you specify the node privilege class, you must also specify either the DOMAIN parameter or the NODE parameter, but not both.

**AUTHority**

Specifies the authority level of a user with node privilege. This parameter is optional.

If an administrator already has system or policy privilege to the policy domain to which the node belongs, this command will not change the administrator's privilege. Possible authority levels are:

**Access**

Specifies that you want to grant client access authority to a user with the node privilege class. This is the default when CLASSES=NODE is specified. A user with client access authority can access a web backup-archive client and perform backup and restore actions on that client.

## GRANT AUTHORITY

**Attention:** A user with client access authority cannot access that client from another system by using the -NODENAME or -VIRTUALNODENAME parameter.

A client node can set the REVOKEREMOTEACCESS option to restrict a user that has node privilege with client access authority from accessing a client workstation that is running a web client. This option does not apply to administrators with client owner authority, system privilege, or policy privilege to the policy domain to which the node belongs.

### Owner

Specifies that you want to grant client owner authority to a user with the node privilege class. A user with client owner authority can access a web backup-archive client through the web client interface and also access their data from another client using the -NODENAME or -VIRTUALNODENAME parameter.

### DOmains

Specifies that you want to grant to the administrator client access or client owner authority to all clients in the specified policy domain. You cannot use this parameter together with the NODE parameter.

### NDe

Specifies that you want to grant the administrator client access or client owner authority to the node. You cannot use this parameter together with the DOMAIN parameter.

### DOmains

Specifies that you want to grant restricted policy privilege to an administrator.

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 wildcard characters to specify a name. Authority for all matching policy domains is granted.

### STGpools

Specifies that you want to grant restricted storage privilege to an administrator. If the STGPOOLS parameter is specified, then CLASSES=STORAGE is optional.

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 wildcard characters to specify a name. Authority for all matching storage pools is granted.

## Example: Grant system privilege to an administrator

Grant system privilege to administrator Larry.

```
grant authority larry classes=system
```



**Example: Grant access to additional policy domains**

Specify additional policy domains that the restricted policy administrator CLAUDIA can manage.

```
grant authority claudia domains=employee_records,prog1
```

**Example: Provide an administrator with unrestricted storage privilege and restricted policy privilege**

Provide administrator TOM with unrestricted storage privilege and restricted policy privilege for the domains whose names start with EMP.

```
grant authority tom classes=storage domains=emp*
```

**Example: Grant an administrator authority restricted to a specific node**

Grant node privilege to user HELP so that help desk personnel can assist the client node LABCLIENT in backing up or restoring data without having other higher-level Tivoli Storage Manager privileges.

```
grant authority help classes=node node=labclient
```

**Related commands**

*Table 164. Commands related to GRANT AUTHORITY*

| Command          | Description  |
|------------------|--|
| QUERY ADMIN      | Displays information about one or more IBM Tivoli Storage Manager administrators.              |
| REVOKE AUTHORITY | Revokes one or more privilege classes or restricts access to policy domains and storage pools. |

## GRANT PROXYNODE (Grant proxy authority to a client node)

Use this command to grant proxy authority to a client node on the Tivoli Storage Manager server.

Target client nodes own the data and agent nodes act on behalf of the target nodes. When granted proxy authority to a target client node, an agent node can perform backup and restore operations for the target node. Data that the agent node stores on behalf of the target node is stored under the target node's name in server storage.

### Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege

### Syntax

```
►► GRANT PROXynode TArget==target_node_name—AGent==agent_node_name—►►
```

### Parameters

#### TArget (Required)

Specifies the name of the node that owns the data. Wildcard names cannot be used to specify the target node name.

#### AGent (Required)

Specifies the name of the node performing operations for the target node. The agent node does not have to be in the same domain as the target node. Wildcard characters and comma-separated lists of node names are allowed.

### Example: Grant proxy authority to a client node

Assume that MOE and JOE are agent nodes in a NAS cluster and are used to backup and restore shared NAS data. To create a proxy authority relationship for target node NASCLUSTER, issue the following command:

```
grant proxynode target=nascluster agent=moe,joe
```

Issue the following command on agent node MOE to back up NAS cluster data stored on the E: drive. The name of the target node is NASCLUSTER.

```
dsmc -asnode=nascluster incremental e:
```

### Related commands

Table 165. Commands related to GRANT PROXYNODE

| Command          | Description   |
|------------------|---|
| QUERY PROXYNODE  | Display nodes with authority to act as proxy nodes. |
| REVOKE PROXYNODE | Revoke proxy authority from an agent node.          |

## HALT (Shut down the server)

Use this command to shut down the server. The HALT command forces an abrupt shutdown, which cancels all the administrative and client node sessions even if they are not completed.

Any transactions in progress interrupted by the HALT command are rolled back when you restart the server. Use the HALT command only after the administrative and client node sessions are completed or canceled. To shut down the server without severely impacting administrative and client node sessions, perform the following steps:

1. Use the DISABLE SESSIONS command to prevent starting new client node sessions.
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 Tivoli Storage Manager).
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.

### Tip:

The HALT command can be replicated using the ALIASHALT server option. Use the server option to define a term other than HALT that performs the same function. The HALT command retains its normal function however, the server option provides an additional method for issuing the HALT command. See “ALIASHALT” on page 1184 for additional information.

## Privilege class

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

## Syntax

▶▶—HALT—◀◀

## Parameters

None.

## Example: Shut down the server

Shut down the server, either from the server console or from an administrative client. All user activity stops immediately and no new activity can start.

```
halt
```

## Related commands

Table 166. Commands related to HALT

| Command        | Description                          |
|----------------|--------------------------------------|
| CANCEL PROCESS | Cancels a background server process. |

## HALT

*Table 166. Commands related to HALT (continued)*

| Command          | Description  |
|------------------|--|
| CANCEL SESSION   | Cancels active sessions with the server.   |
| DISABLE SESSIONS | Prevents new sessions from accessing IBM Tivoli Storage Manager but permits existing sessions to continue. |
| ENABLE SESSIONS  | Resumes server activity following the DISABLE command or the ACCEPT DATE command.                          |
| QUERY PROCESS    | Displays information about background processes.   |
| QUERY SESSION    | Displays information about all active administrator and client sessions with IBM Tivoli Storage Manager.   |

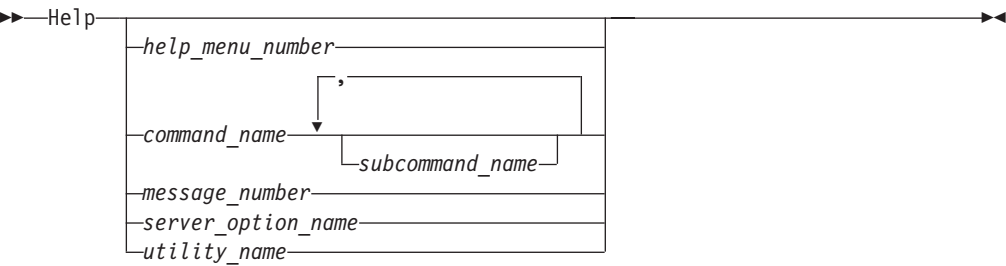
# HELP (Get help on commands and error messages)

Use this command to display administrative commands and error messages. You can issue the command from an administrative command line client.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

*help\_topic\_number*  
Specifies the number of your selection from the help topics. This parameter is optional.

*command\_name*  
Specifies the name of the administrative command you want to display. This parameter is optional.

*subcommand\_name*  
Specifies up to two of the subcommand names that are associated with the name of the administrative command that you want to display. This parameter is optional.

*message\_number*  
Specifies the number of the 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 ANE or ANS). Do not include the prefix and severity code when specifying an error message number.

*server\_option\_name*  
Specifies the name of the server option for which you want to display information. This parameter is optional.

*utility\_name*  
Specifies the name of the server utility for which you want to display information. This parameter is optional.

## Example: Display the help topics

Display the help topics for the command-line interface.

help

Partial output:

```
1.0 Administering the server from the command line
  1.1 Issuing commands from the administrative client
    1.1.1 Starting and stopping the administrative client
    1.1.2 Monitoring server activities from the administrative client
```

### Example: Display a help topic by using the help topic number

Display help information by using the help topic number. Topic 1.1.2 contains information on commands that can be issued by any administrator.

```
help 1.1.2
```

### Example: Display help for one command

Display help information on the REMOVE commands.

```
help remove
```

```
3.44 REMOVE commands
Use the REMOVE commands to remove an object.
The following is a list of REMOVE commands:
* 3.44.1, "REMOVE ADMIN (Delete an administrator)"
* 3.44.2, "REMOVE NODE (Delete a node or an associated machine node)"
```

### Example: Display help for a specific error message

Display help information on the error message ANR2535E.

```
help 2535
```

```
ANR2535E Command: The node node name cannot be removed or renamed
           because it has an associated data mover.
Explanation: You attempted to remove or rename a node that has an
associated data mover.
System action: The server does not remove or rename the node.
User response: To remove or rename the node, delete the associated data
mover and reissue the command.
```

### Example: Display help for a specific option

Display the description, syntax, and an example for the COMMMETHOD server option.

```
help commethod
```

### Example: Display help for a specific utility

Display the description, syntax, and an example for the DSMSERV utility.

```
help dsmserv
```

### Related commands

None.

---

## IDENTIFY DUPLICATES (Identify duplicate data in a storage pool)

Use this command to start or stop processes that identify duplicate data in a storage pool. You can specify the number of duplicate-identification processes and their duration.

When you create a new storage pool for deduplication, you can specify 0 - 20 duplicate-identification processes. Tivoli Storage Manager starts the specified number of duplicate-identification processes automatically when the server is started. If you do not stop them, they run indefinitely.

With the IDENTIFY DUPLICATES command, you can start additional processes, stop some or all of the processes, and specify an amount of time that the change remains in effect. If you increased or decreased the number of duplicate-identification processes, you can use the IDENTIFY DUPLICATES command to reset the number of processes to the number that is specified in the storage pool definition.

If you did not specify any duplicate-identification processes in the storage pool definition, you can use the IDENTIFY DUPLICATES command to start and stop all processes manually.

This command starts or stops a background process or processes that you can cancel with the CANCEL PROCESS command. To display information about background processes, use the QUERY PROCESS command.

### Important:

- You can also change the number of duplicate-identification processes by updating the storage pool definition using the UPDATE STGPOOL command. However, when you update a storage pool definition, you cannot specify a duration. The processes that you specify in the storage pool definition run indefinitely, or until you issue the IDENTIFY DUPLICATES command, update the storage pool definition again, or cancel a process.

Issuing the IDENTIFY DUPLICATES does not change the setting for the number of duplicate-identification processes in the storage pool definition.

- Duplicate-identification processes can be either active or idle. Processes that are deduplicating files are active. Processes that are waiting for files to deduplicate are idle. Processes remain idle until volumes with data to be deduplicated become available. Processes stop only when cancelled or when you change the number of duplicate-identification processes for the storage pool to a value less than that currently specified. Before a duplicate-identification process stops, it must finish the file that it is deduplicating.

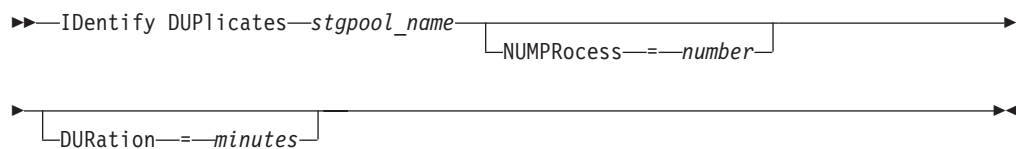
The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.

### Privilege class

To issue this command, you must have system privilege.

## IDENTIFY DUPLICATES

### Syntax



### Parameters

#### *stgpool\_name* (Required)

Specifies the storage pool name in which duplicate data is to be identified. You can use wildcards.

#### NUMProcess

Specifies the number of duplicate-identification processes to run after the command completes. You can specify 0 - 20 processes. The value that you specify for this parameter overrides the value that you specified in the storage pool definition or the most recent value that was specified when you last issued this command. If you specify zero, all duplicate-identification processes stop.

This parameter is optional. If you do not specify a value, the Tivoli Storage Manager server starts or stops duplicate-identification processes so that the number of processes is the same as the number that is specified in the storage pool definition.

For example, suppose you define a new storage pool and specify two duplicate-identification processes. Later, you issue the IDENTIFY DUPLICATES command to increase the number of processes to four. When you issue the IDENTIFY DUPLICATES command again without specifying a value for the **NUMPROCESS** parameter, the server stops two duplicate-identification processes.

If you specified 0 processes when you defined the storage pool definition and you issue IDENTIFY DUPLICATES without specifying a value for **NUMPROCESS**, any running duplicate-identification processes stop, and the server does not start any new processes.

**Remember:** When you issue IDENTIFY DUPLICATES without specifying a value for **NUMPROCESS**, the **DURATION** parameter is not available. Duplicate-identification processes specified in the storage pool definition run indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.

When the server stops a duplicate-identification process, the process completes the current physical file and then stops. As a result, it might take several minutes to reach the number of duplicate-identification processes that you specified as a value for this parameter.

#### DURation

Specifies the maximum number of minutes (1 - 9999) that this command remains in effect. At the end of the specified time, the Tivoli Storage Manager server starts or stops duplicate-identification processes so that the number of processes is the same as the number that is specified in the storage pool definition.



This parameter is optional. If you do not specify a value, the processes that are running after the command is issued will run indefinitely. They will end only if you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.

For example, if you define a storage pool with two duplicate-identification processes and you issue the IDENTIFY DUPLICATES command with DURATION=60 and NUMPROCESS=4, the server starts two additional duplicate-identification processes that run for 60 minutes. At the end of that time, two processes finish the files they are working on and stop. The two processes that stop might not be the same two processes that started as a result of issuing this command.

The Tivoli Storage Manager server stops idle processes first. If, after stopping all idle processes, more processes need to be stopped, the Tivoli Storage Manager server notifies active processes to stop.

When the server stops a duplicate-identification process, the process completes the current physical file and then stops. As a result, it might take several minutes to reach the amount of time that you specified as a value for this parameter.

## Example: Controlling the number and duration of duplicate-identification processes

In this example, you specified three duplicate-identification processes in the storage pool definition. You use the IDENTIFY DUPLICATES command to change the number of processes and to specify the amount of time the change is to remain in effect.

Table 167. Controlling duplicate-identification processes manually

| The storage pool definition specifies three duplicate-identification processes. Using the IDENTIFY DUPLICATES command, you specify... | ...and a duration of... | The result is...  |
|---|-------------------------|---|
| 2 duplicate-identification processes  | None specified          | One duplicate-identification processes finishes the file it is working on, if any, and then stops. Two processes run indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.                       |
|   | 60 minutes              | One duplicate-identification process finishes the file it is working on, if any, and then stops. After 60 minutes, the server starts one process so that three are running.   |
| 4 duplicate-identification processes  | None specified          | The server starts one duplicate-identification process. Four processes run indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.   |
|   | 60 minutes              | The server starts one duplicate-identification process. At the end of 60 minutes, one process finishes the file it is working on, if any, and then stops. The additional process started by this command might not be the one that stops when the duration has expired. |

## IDENTIFY DUPLICATES

Table 167. Controlling duplicate-identification processes manually (continued)

| The storage pool definition specifies three duplicate-identification processes. Using the IDENTIFY DUPLICATES command, you specify... | ...and a duration of... | The result is...  |
|---|-------------------------|---|
| 0 duplicate-identification processes  | None specified          | All duplicate-identification processes finish the files that they are working on, if any, and stop. This change lasts indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process.                          |
|   | 60 minutes              | All duplicate-identification processes finish the files that they are working on, if any, and stop. At the end of 60 minutes, the server starts three processes.  |
| None specified  | Not available           | The number of duplicate-identification processes resets to the number of processes specified in the storage pool definition. This change lasts indefinitely, or until you reissue the IDENTIFY DUPLICATES command, update the storage pool definition, or cancel a process. |

### Example: Identify duplicates in a storage pool

Identify duplicates in a storage pool, STGPOOLA, using three duplicate-identification processes. Specify that this change is to remain in effect for 60 minutes.

```
identify duplicates stgpoola duration=60 numprocess=3
```

### Related commands

Table 168. Commands related to IDENTIFY DUPLICATES

| Command        | Description   |
|----------------|---|
| CANCEL PROCESS | Cancels a background server process.                                  |
| DEFINE STGPOOL | Defines a storage pool as a named collection of server storage media. |
| QUERY PROCESS  | Displays information about background processes.                      |
| QUERY STGPOOL  | Displays information about storage pools.                             |
| UPDATE STGPOOL | Changes the attributes of a storage pool.                             |

---

## IMPORT commands

Use the IMPORT commands to import information from export media to a Tivoli Storage Manager server.

The following list shows the IMPORT commands for Tivoli Storage Manager:

- “IMPORT ADMIN (Import administrator information)” on page 502
- “IMPORT NODE (Import client node information)” on page 505
- “IMPORT POLICY (Import policy information)” on page 511
- “IMPORT SERVER (Import server information)” on page 514

### IMPORT ADMIN (Import administrator information)

Use this command to import administrator and authority definitions for one or more administrators from export media to the Tivoli Storage Manager server.

You can use the `QUERY ACTLOG` command to view the status of the import operation.

You can also view this information from the server console.

This command generates a background process that can be cancelled with the CANCEL PROCESS command. If an IMPORT ADMIN background process is cancelled, some of the data is already imported. To display information on background processes, use the QUERY PROCESS command.

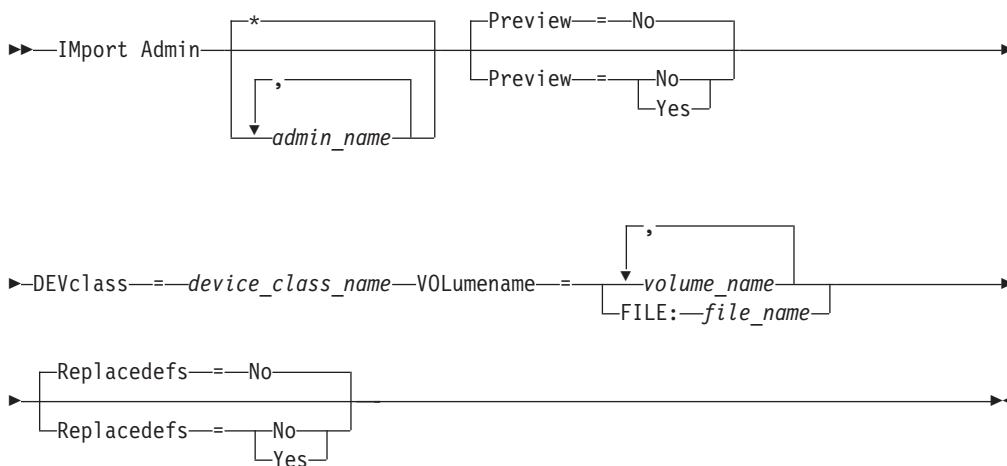
**Restriction:**

- If target and source server levels are not compatible, the operation may not work. See the *Administrator's Guide* for server compatibility requirements.
- If the administrator definition being imported includes analyst authority, the administrator definition is imported but not the analyst authority. Analyst authority is not valid for servers at V6.1 or later.
- Importing data from a CENTERA device class is not supported. However, files being imported can be stored on a CENTERA storage device.

## Privilege class

To issue this command, you must have system privilege.

## Syntax



## Parameters

*admin name*

Specifies the administrators for which you want to import information. This parameter is optional. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names.

## Preview

Specifies whether you want to preview the results of the import operation,

without importing administrator information. If you specify YES for the value, you must mount the export volumes. This parameter is optional. The default value is NO. The values are:

**No**

Specifies that the information is to be imported.

**Yes**

Specifies that the operation will be previewed but not performed. 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 (Required)**

Specifies the device class from which import data is to be read.

You cannot specify the DISK, NAS, or CENTERA device classes. If all drives for the device class are busy when the import runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

**VOLumentname (Required)**

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. The values are:

*volume\_name*

Specifies the volume name. To specify multiple volumes, separate names with commas and no intervening spaces.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes used for the imported data. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify  |
|-----------------|--|
| Tape            | 1 – 6 alphanumeric characters.   |
| FILE            | Any fully qualified file name string. For example:<br><br>/imdata/mt1. |
| OPTICAL         | 1–32 alphanumeric characters.  |
| REMOVABLEFILE   | 1–6 alphanumeric characters.   |
| SERVER          | 1–250 alphanumeric characters.   |

**Replacedefs**

Specifies whether to replace administrator definitions on the target server. The default value is NO. The values are:

**No**

Specifies that definitions are not to be replaced.

**Yes**

Specifies that definitions are to be replaced.

### Example: Import administrator information from specific tape volumes

From the server, import the information for all defined administrators from tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class. Issue the command:

```
import admin devclass=menu1
volumenames=tape01,tape02,tape03
```

### Example: Import administrator information from tape volumes listed in a file

From the server, import the information for all defined administrators from tape volumes that are listed in the following file:

```
TAPEVOL
```

This file contains these lines:

```
TAPE01
TAPE02
TAPE03
```

Specify that these tape volumes be read by a device assigned to the MENU1 device class. Issue the command:

```
import admin devclass=menu1 volumenames=file:tapevol
```

### Related commands

Table 169. Commands related to IMPORT ADMIN

| Command        | Description   |
|----------------|---|
| CANCEL PROCESS | Cancels a background server process.                    |
| EXPORT ADMIN   | Copies administrative information to external media.    |
| IMPORT NODE    | Restores client node information from external media.   |
| IMPORT POLICY  | Restores policy information from external media.        |
| IMPORT SERVER  | Restores all or part of the server from external media. |
| QUERY ACTLOG   | Displays messages from the server activity log.         |
| QUERY PROCESS  | Displays information about background processes.        |

## IMPORT NODE (Import client node information)

Use this command to import client node definitions from a server or sequential media to a target IBM Tivoli Storage Manager server.

The imported nodes are associated with the same policy domain as they were on the source server, if you specify the domain and if that policy domain exists on the target server. Otherwise, imported nodes are associated with the STANDARD policy domain on the target server.

IBM Tivoli Storage Manager servers with retention protection enabled do not allow import operations.

### Restriction:

1. If target and source server levels are not compatible, the operation may not work. See the *Administrator's Guide* for server compatibility requirements.
2. Importing data from a CENTERA device class is not supported. However, files being imported can be stored on a CENTERA storage device.

You can use the QUERY ACTLOG command to view the status of the import operation. You can also view this information from the server console.

This command generates a background process that can be cancelled with the CANCEL PROCESS command. If an IMPORT NODE background process is cancelled, some of the data may already be imported. To display information on background processes, use the QUERY PROCESS command.

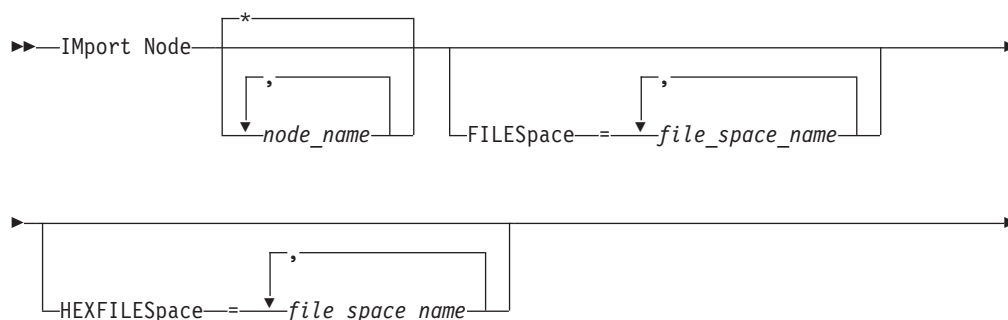
For a server that has clients with support for Unicode, you may need to have the server convert the file space name that you enter, or use the following parameter:

- **HEXFILESPACE** parameter
- **UNIFILESPACE** parameter

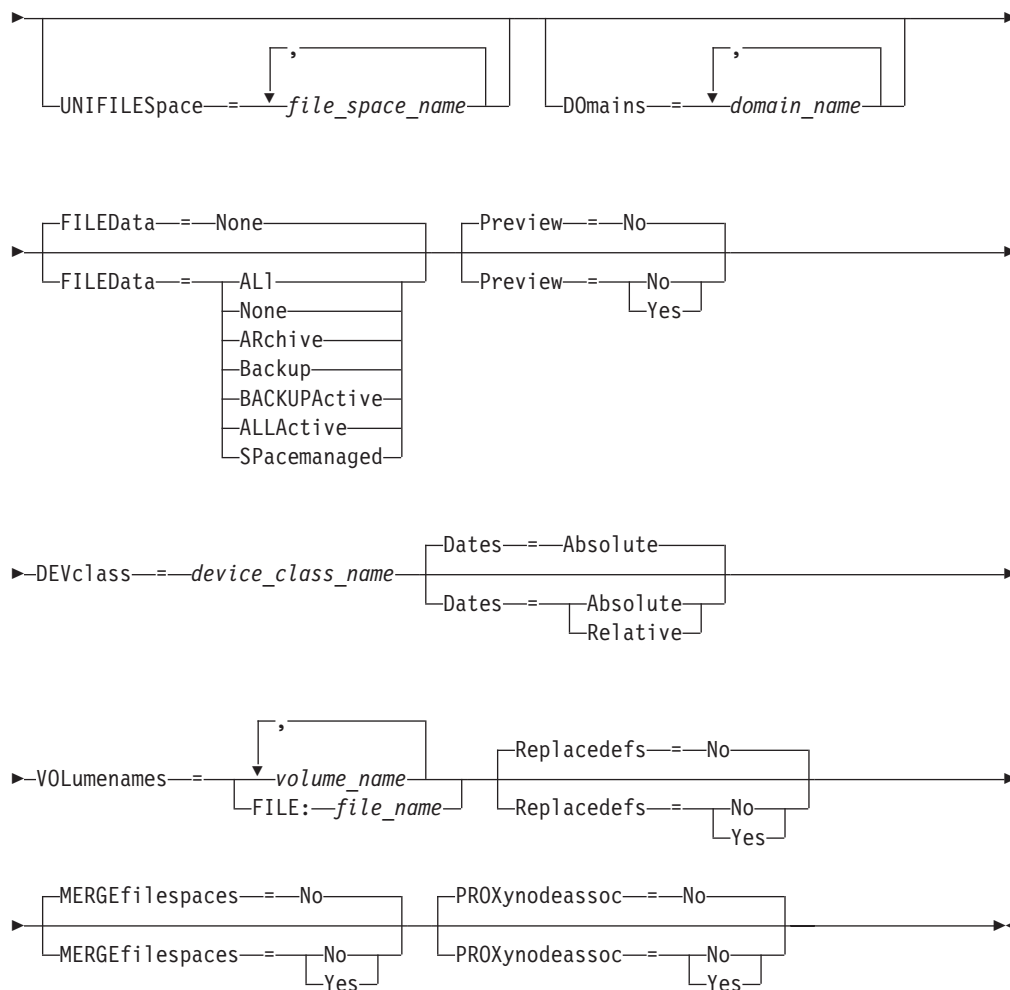
### Privilege class

To issue this command, you must have system privilege.

### Syntax



## IMPORT NODE



## Parameters

### *node\_name*

Specifies the client nodes for which you want to import information. This parameter is optional.

Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. All matching nodes are included in the list.

### **FILESpace**

Specifies file space names for which you want to import information. This parameter is optional. The default is all file spaces.

Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names.

### **Important:**

1. Existing file spaces are not replaced. New file spaces are created when identical names are encountered. However, this new name may match an existing name on the client node, which may have file spaces not yet backed up to the server.



2. This parameter can only be specified for non-Unicode file spaces. To import all file spaces, that are both Unicode and non-Unicode, use the **FILEDATA=ALL** parameter without the **FILESPACE** and **UNIFILESPACE** parameters.

### **DOmains**

Specifies the policy domains from which to import node information. These domains must have been included in the data that was exported. This parameter is optional. The default is all domains that were exported.

Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify a name.

### **FILEData**

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**.

If you are importing from sequential media, 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, two drives are needed to import the node information. The mount limit for the device class must be 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. The values are:

#### **AlI**

The server imports all backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client. This includes both Unicode and non-Unicode file spaces.

#### **None**

Only node definitions are imported. The server does not import any files.

#### **ARchive**

The server imports only archived files.

#### **Backup**

The server imports only backup versions, whether active or inactive.

#### **BACKUPActive**

The server imports only active backup versions. These active backup versions are the active versions in the Tivoli Storage Manager database at the time that the **IMPORT** command is issued.

#### **ALLActive**

The server imports all active backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client. The active backup versions are the active versions in the Tivoli Storage Manager database at the time that the **IMPORT** command is issued.

#### **SPacemanaged**

The server imports only files that were migrated by a Tivoli Storage Manager for Space Management client.

### **Preview**

Specifies whether to preview the results of the import operation, without

## IMPORT NODE

importing information. The PREVIEW=YES option requires that you mount the export volumes. 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 importing files. Information is reported to the server console and the activity log.

### DEVclass (Required)

Specifies the device class from which import data is to be read. You cannot specify the DISK, NAS, or CENTERA device classes.

If all drives for the device class are busy when the import runs, the server cancels lower priority operations, such as reclamation, to make a drive available.

### Dates

Specifies whether the date of the file copies are as specified when the files were exported (ABSOLUTE), or are adjusted 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 expiration 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 expired.

For example, assume that an export tape contains an archive file copy that was archived 5 days before the export operation. If the media are 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 expiration depending on 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. 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

The dates for file copies are set to the values specified when the files were exported.

### Relative

The dates for file copies are adjusted to the date of import.

### VOLumenames (Required)

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. The values are:

*volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes used for the

imported data. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify  |
|-----------------|--|
| Tape            | 1 – 6 alphanumeric characters.                                   |
| FILE            | Any fully qualified file name string. An example is /imdata/mt1. |
| OPTICAL         | 1–32 alphanumeric characters.                                    |
| REMOVABLEFILE   | 1–6 alphanumeric characters.                                     |
| SERVER          | 1–250 alphanumeric characters.                                   |

### Replacedefs

Specifies whether to replace definitions on the target server. The default value is NO. The values are:

#### No

Objects are not to be replaced.

#### Yes

Objects are to be replaced.

### HEXFILESpace

Specifies the hexadecimal representation of the file space names in UTF-8 format. Separate multiple names with commas and no intervening spaces. This parameter is optional.

To view the hexadecimal representation of a file space name, you can use the QUERY FILESPACE command with FORMAT=DETAILED.

### UNIFILESpace

Specifies the file spaces that are known to the server to be Unicode enabled. The server converts the names you enter from the server code page to the UTF-8 code page to find the file spaces to import. The success of the conversion depends on the actual characters in the name and the server's code page. Separate multiple names with commas and no intervening spaces. This parameter is optional.

### MERGEfilespace

Specifies whether Tivoli Storage Manager merges client files into existing file spaces on the target server (if they exist), or if Tivoli Storage Manager generates new file space names. The default is NO.

Valid values are:

#### Yes

Specifies that imported data on the target server is merged with the existing file space, if a file space with the same name already exists on the target server.

#### No

Specifies that Tivoli Storage Manager generates a new file space name for imported data on the target server if file spaces with the same name already exist.

## IMPORT NODE

### PROXynodeassoc

Specifies if proxy node associations are imported. This parameter is optional. The default value is NO.

### Example: Import client node information from tapes

From the server, import client node information from tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class.

```
import node devclass=menu1 volumenames=tape01,tape02,tape03
```

### Example: Import client node information from tapes listed in a file

From the server, import client node information from tape volumes listed in a file named TAPEVOL.

This file contains these lines:

```
TAPE01  
TAPE02  
TAPE03
```

Specify that these tape volumes be read by a device assigned to the MENU1 device class.

```
import node devclass=menu1 volumenames=file:tapevol
```

### Example: Import the active backup for a client node

From the server, import the active backup versions of file data for client node JOE from tape volume TAPE01. The file space is Unicode.

```
import node joe unifilespace=\\joe\c$ filedata=backupactive devclass=menu1  
volumenames=tape01
```

## Related commands

Table 170. Commands related to IMPORT NODE

| Command           | Description  |
|-------------------|--|
| CANCEL PROCESS    | Cancels a background server process.                     |
| COPY ACTIVATEDATA | Copies active backup data.                               |
| EXPORT NODE       | Copies client node information to external media.        |
| IMPORT ADMIN      | Restores administrative information from external media. |
| IMPORT POLICY     | Restores policy information from external media.         |
| IMPORT SERVER     | Restores all or part of the server from external media.  |
| QUERY ACTLOG      | Displays messages from the server activity log.          |
| QUERY PROCESS     | Displays information about background processes.         |

## IMPORT POLICY (Import policy information)

Use this command to import policy domain information from sequential export media to the Tivoli Storage Manager server. IBM Tivoli Storage Manager servers with retention protection enabled do not allow import operations.

Tivoli Storage Manager client data can be moved between servers with export and import processing, if the same removable media type is supported on both platforms.

### Restriction:

1. If target and source server levels are not compatible, the operation may not work. See the *Administrator's Guide* for server compatibility requirements.
2. Importing data from a CENTERA device class is not supported. However, files being imported can be stored on a CENTERA storage device.

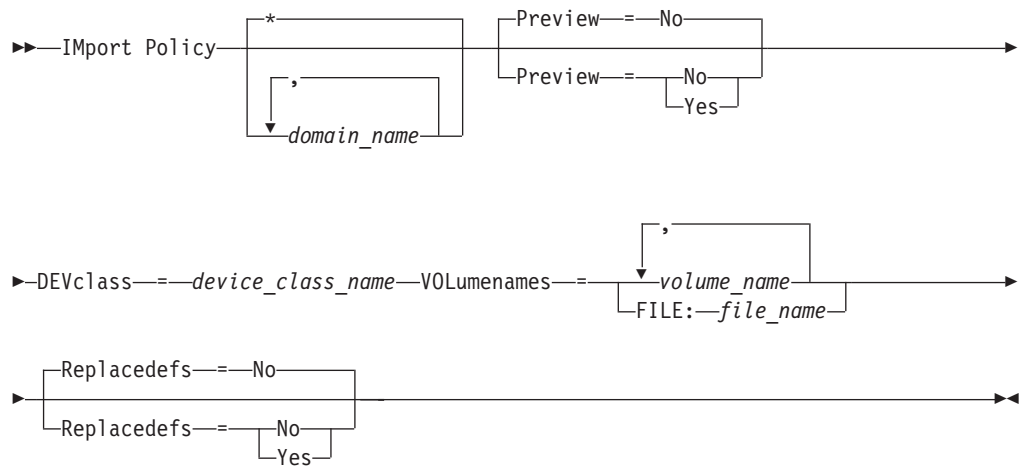
You can use the QUERY ACTLOG command to view the status of the import operation. You can also view this information from the server console.

This command generates a background process that can be cancelled with the CANCEL PROCESS command. If an IMPORT POLICY background process is cancelled, 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

#### domain\_name

Specifies the policy domains for which information is to be imported. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. The default (\*) is all policy.

#### Preview

Specifies whether you want to preview the results of the import operation

## IMPORT POLICY

without importing information. The PREVIEW=YES option requires that you mount the export volumes. This parameter is optional. The default value is NO. The values are:

### No

Specifies that the information is to be imported.

### Yes

Specifies that the operation will be previewed but not performed. Information is reported to the server console and the activity log.

### DEVclass (Required)

Specifies the device class from which import data is to be read. You cannot specify the DISK device class.

You also cannot specify the NAS or CENTERA device classes.

If all drives for the device class are busy when the import runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

### VOLumenames (Required)

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. The values are:

*volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify   |
|-----------------|---|
| Tape            | 1 – 6 alphanumeric characters.                                    |
| FILE            | Any fully qualified file name string. For example:<br>/imdata/mt1 |
| OPTICAL         | 1–32 alphanumeric characters.                                     |
| REMOVABLEFILE   | 1–6 alphanumeric characters.                                      |
| SERVER          | 1–250 alphanumeric characters.                                    |

### Replacedefs

Specifies whether to replace policy definitions on the target server. The default value is NO. The 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.

### Example: Import policy information from specific tape volumes

From the server, import the information for all defined policies from tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class.

```
import policy devclass=menu1
volumenames=tape01,tape02,tape03
```

### Example: Import policy information from tape volumes listed in a file

From the server, import the information for all defined policies from tape volumes listed in a file named thus:

```
TAPEVOL
TAPEVOL.DATA
```

Specify that these tape volumes be read by a device assigned to the MENU1 device class. The file contains the following lines:

```
TAPE01
TAPE02
TAPE03
import policy devclass=menu1 volumenames=file:tapevol
```

### Related commands

Table 171. Commands related to IMPORT POLICY

| Command        | Description  |
|----------------|--|
| CANCEL PROCESS | Cancels a background server process.                     |
| EXPORT POLICY  | Copies policy information to external media.             |
| IMPORT ADMIN   | Restores administrative information from external media. |
| IMPORT NODE    | Restores client node information from external media.    |
| IMPORT SERVER  | Restores all or part of the server from external media.  |
| QUERY ACTLOG   | Displays messages from the server activity log.          |
| QUERY PROCESS  | Displays information about background processes.         |

## IMPORT SERVER (Import server information)

Use this command to copy all or part of the server control information and client file data (if specified) from export media to the Tivoli Storage Manager server.

IBM Tivoli Storage Manager servers with retention protection enabled do not allow import operations.

### Restriction:

- If target and source server levels are not compatible, the operation might not work.

For server compatibility requirements, see the *Administrator's Guide*.

- Importing data from a CENTERA device class is not supported. However, files being imported can be stored on a CENTERA storage device.

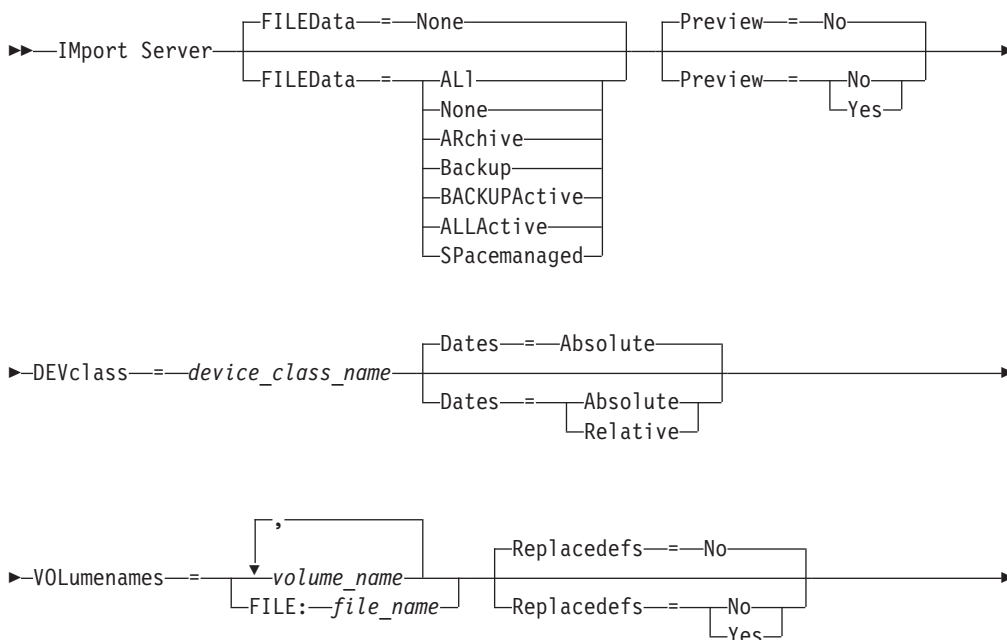
You can also initiate an import of server information and client file data directly from the originating server. For more information see the EXPORT commands.

This command generates a background process that can be cancelled with the CANCEL PROCESS command. If an IMPORT SERVER background process is cancelled, some of the data is already imported. To display information on background processes, use the QUERY PROCESS command.

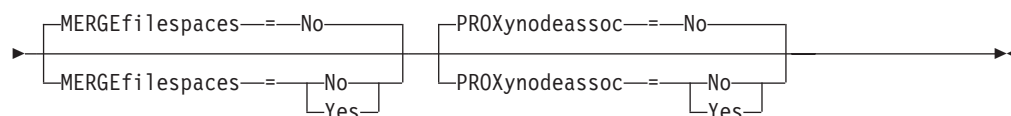
### Privilege class

To issue this command, you must have system privilege.

### Syntax







## Parameters

### FILEData

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 to access the file data is determined by the device class for the storage pool. If it is the same device class specified in this command, two drives are needed to import information. The mount limit for the device class must 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. The values are:

#### ALL

Tivoli Storage Manager imports all backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client.

#### None

Tivoli Storage Manager does not import files, only node definitions.

#### ARchive

Tivoli Storage Manager imports only archived files.

#### Backup

Tivoli Storage Manager imports only backup versions, whether the versions are active or inactive.

#### BACKUPActive

Tivoli Storage Manager imports only active backup versions. These active backup versions are the active versions in the Tivoli Storage Manager database at the time that the IMPORT command is issued.

#### ALLActive

Tivoli Storage Manager imports all active backup versions of files, all archived files, and all files that were migrated by a Tivoli Storage Manager for Space Management client. The active backup versions are the active versions in the Tivoli Storage Manager database at the time that the IMPORT command is issued.

#### SPacemanaged

Tivoli Storage Manager imports only files that were migrated by a Tivoli Storage Manager for Space Management client.

### Preview

Specifies whether to preview the results of the import operation, without importing information. The PREVIEW=YES option still requires that you mount the export volumes. 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 the operation will be previewed but not performed.  
Information is transferred to the server console and the activity log.

### DEVclass (Required)

Specifies the device class from which import data is to be read. You cannot specify the DISK, NAS, or CENTERA device classes.

If all drives for the device class are busy when the import runs, Tivoli Storage Manager cancels lower priority operations, such as reclamation, to make a drive available.

### Dates

Specifies whether the dates of the file copies are set as specified when the files were exported (ABSOLUTE), or are adjusted to the date of import (RELATIVE). The default value is ABSOLUTE.

If the import 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 expiration 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 expired.

For example, assume that an import tape contains an archive file copy that was archived 5 days before the export operation. If the export media are 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 expiration depending upon the retention value that is specified in the file's management class. Specifying DATES=RELATIVE results in resetting the archive date for the file to 5 days ago during import. 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

The dates for file copies are set to the values specified when the files were exported.

#### Relative

The date for file copies are adjusted to the date of import.

### VOLumenames (Required)

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. The values are:

#### *volume\_name*

Specifies the volume name. To specify multiple volumes, separate the names with commas and no intervening spaces.

#### **FILE:***file\_name*

Specifies the name of a file that contains a list of volumes used for the imported data. In the file, each volume name must be on a separate line. Blank and comment lines that begin with an asterisk are ignored.

Use these naming conventions when specifying volumes associated with the following device types:

| For this device | Specify                        |
|-----------------|--------------------------------|
| Tape            | 1 – 6 alphanumeric characters. |

| For this device | Specify  |
|-----------------|--|
| FILE            | Any fully qualified file name string. An example is /imdata/mt1. |
| OPTICAL         | 1–32 alphanumeric characters.                                    |
| REMOVABLEFILE   | 1–6 alphanumeric characters.                                     |
| SERVER          | 1–250 alphanumeric characters.                                   |

**Replacedefs**

Specifies whether to replace objects on the server. 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.

**Yes**

Specifies that objects are to be replaced by the imported objects.

**MERGEfilespace**

Specifies whether Tivoli Storage Manager merges client files into existing file spaces on the target server (if they exist), or if Tivoli Storage Manager generates new file space names. You cannot merge non-Unicode and Unicode file spaces together. The default is NO. Possible values are:

**No**

Specifies that Tivoli Storage Manager generates a new file space name for imported data on the target server if file spaces with the same name already exist.

**Yes**

Specifies that imported data on the target server is merged with the existing file space, if a file space with the same name already exists on the target server.

**PROXynodeassoc**

Specifies if proxy node associations are imported. This parameter is optional. The default value is NO.

**Example: Import the information for all defined servers from specific tapes**

From the server, import the information for all defined servers from tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class.

```
import server devclass=menu1 volumenames=tape01,tape02,tape03
```

**Example: Import information for all defined servers from specific tapes and specify files are merged into existing file spaces**

From the server, import the information for all defined servers from tape volumes TAPE01, TAPE02, and TAPE03. Specify that these tape volumes be read by a device assigned to the MENU1 device class and that client files be merged into file spaces on the target server if file spaces of the same names already exist.

```
import server devclass=menu1 volumenames=tape01,tape02,tape03 mergefilespace=yes
```

### Example: Import information for all defined servers from tapes listed in a file

From the server, import the information for all defined servers from tape volumes listed in a file named TAPEVOL. Specify that the tape volumes are read by a device assigned to the MENU1 device class. The input file contains these lines:

```
TAPE01
TAPE02
TAPE03
import server devclass=menu1 volumenames=file:tapevol
```

### Related commands

*Table 172. Commands related to IMPORT SERVER*

| Command           | Description  |
|-------------------|--|
| CANCEL PROCESS    | Cancels a background server process.                     |
| COPY ACTIVATEDATA | Copies active backup data.                               |
| EXPORT SERVER     | Copies all or part of the server to external media.      |
| IMPORT ADMIN      | Restores administrative information from external media. |
| IMPORT NODE       | Restores client node information from external media.    |
| IMPORT POLICY     | Restores policy information from external media.         |
| QUERY ACTLOG      | Displays messages from the server activity log.          |
| QUERY PROCESS     | Displays information about background processes.         |

# INSERT MACHINE (Insert machine characteristics information or recovery instructions)

Use this command to add client machine characteristics or recovery instructions to existing machine information in the database.

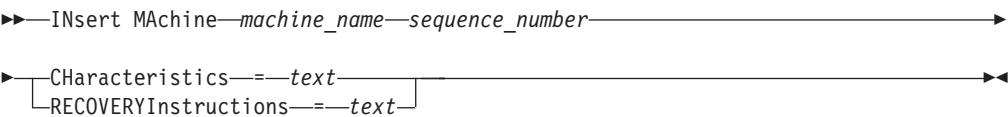
You can write a program to read files containing the information and generate the appropriate INSERT MACHINE commands. See the *Administrator's Guide* for details.

You can use QUERY commands to retrieve the information if a disaster occurs.

## Privilege class

To issue this command, you must have system privilege.

## Syntax



## Parameters

*machine\_name* (Required)

Specifies the name of the client machine.

*sequence\_number* (Required)

Specifies the sequence number for the line of text in the database.

### CHARACTERISTICS

Specifies machine characteristics information. You must specify the characteristics or recovery instructions, but not both. Enclose the text in quotation marks if it contains blank characters. The text can be up to 1024 characters.

### RECOVERYINSTRUCTIONS

Specifies recovery instructions. You must specify the characteristics or recovery instructions, but not both. Enclose the text in quotation marks if it contains blank characters. The text can be up to 1024 characters.

## Example: Update a machine's information

For the machine DISTRICT5, insert this characteristics text on line 1: "Machine owner is Mary Smith".

```
insert machine district5 1
characteristics="Machine owner is Mary Smith"
```

## Related commands

Table 173. Commands related to INSERT MACHINE

| Command        | Description                |
|----------------|----------------------------|
| DEFINE MACHINE | Defines a machine for DRM. |
| DELETE MACHINE | Deletes a machine.         |

## INSERT MACHINE

*Table 173. Commands related to INSERT MACHINE (continued)*

| Command       | Description                          |
|---------------|--------------------------------------|
| QUERY MACHINE | Displays information about machines. |

## ISSUE MESSAGE (Issue a message from a server script)

Use this command with return code processing in a script to issue a message from a server script to determine where the problem is with a command in the script.

### Privilege class

Any administrator can issue this command.

### Syntax

►—ISSUE MESSAGE—*message\_severity*—*message\_text*—◄

### Parameters

#### *message\_severity* (Required)

Specifies the severity of the message. The message severity indicators are:

- I** Information. ANR1496I is displayed in the message text.
- W** Warning. ANR1497W is displayed in the message text.
- E** Error. ANR1498E is displayed in the message text.
- S** Severe. ANR1499S is displayed in the message text.

#### *message\_text* (Required)

Specifies the description of the message.

### Example: Issue a message from a server script

Assume you have a script called `backupscrip` that quiesces a client's database, takes a backup of that database, and then restarts the client's database. For illustration, your script results in a non-zero return code. Use the `ISSUE MESSAGE` command with the message severity and message text. The following is an example of a server script that calls `backupscrip` on the client machine and issues messages based on the return code from `backupscrip`.

```
issue message i "Starting backup"
define clientaction nodename action=command objects="c:\backupscrip" wait=yes
if (101) goto qfail
if (102) goto qwarn
if (103) goto backupf
if (104) goto restartf
issue message i "Backup of database complete"
exit
qfail: issue message e "Quiesce of database failed"
exit
qwarn: issue message w "Quiesce of database failed, taking fuzzy backup"

exit
backupf: issue message e "Backup of database failed"
exit
restartf: issue message s "Database restart failed"
exit
```

### Command

```
issue message e "quiesce of database failed"
```

### Related commands

*Table 174. Commands related to ISSUE MESSAGE*

| Command       | Description  |
|---------------|--|
| COPY SCRIPT   | Creates a copy of a script.                                |
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| DELETE SCRIPT | Deletes the script or individual lines from the script.    |
| RENAME SCRIPT | Renames a script to a new name.                            |
| RUN           | Runs a script.   |
| UPDATE SCRIPT | Changes or adds lines to a script.                         |



## LABEL LIBVOLUME (Label a library volume)

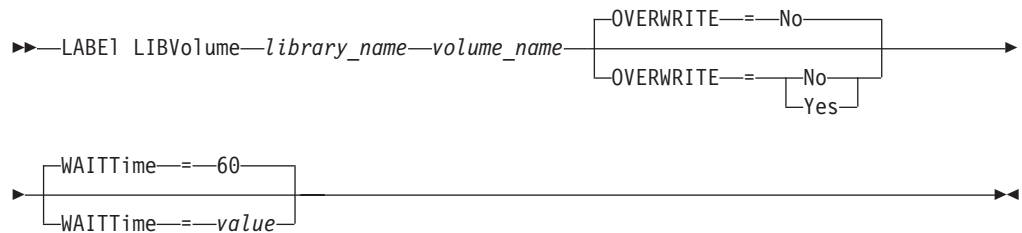
Use this command to label tape volumes or, in an automated library, to label the volumes automatically as they are checked in. With this command, the server uses the full-length label with which the volumes are often prelabeled.

**Important:** Use this command only for MANUAL, SCSI, ACSLS, and 349X libraries. The command processing will not wait for a drive to become available, even if the drive is only in the IDLE state. If necessary, you can make a library drive available by issuing the DISMOUNT VOLUME command to dismount the volume in that particular drive. When the library drive becomes available, you can reissue the LABEL LIBVOLUME command.

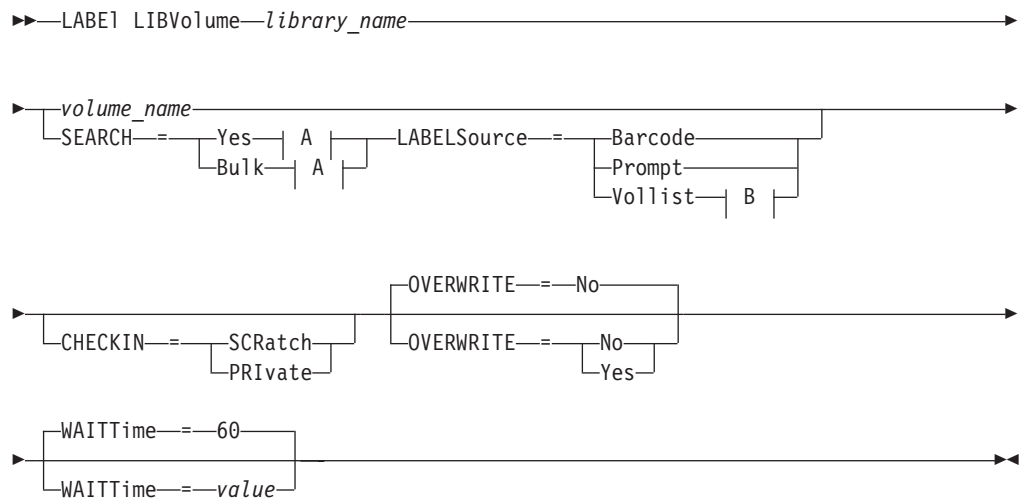
### Privilege class

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

### Syntax for a manual library

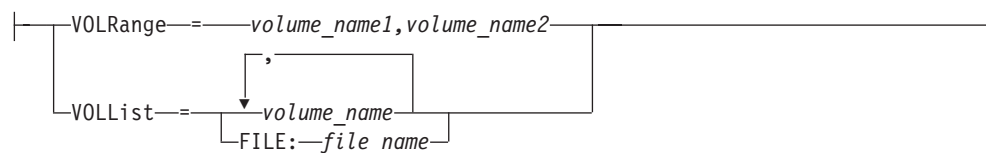


### Syntax for a SCSI library



## LABEL LIBVOLUME

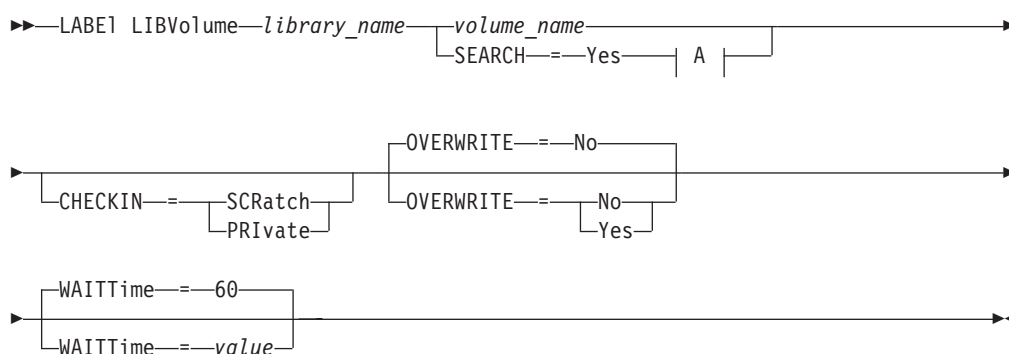
### A (SEARCH=Yes, SEARCH=Bulk):



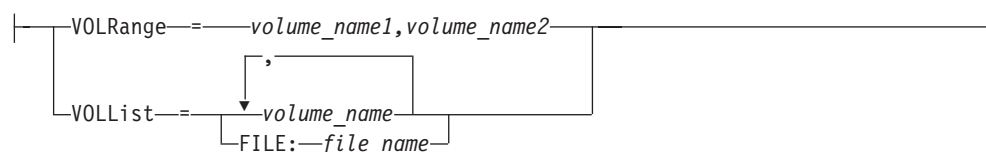
### B (LABELSource=Vollist):



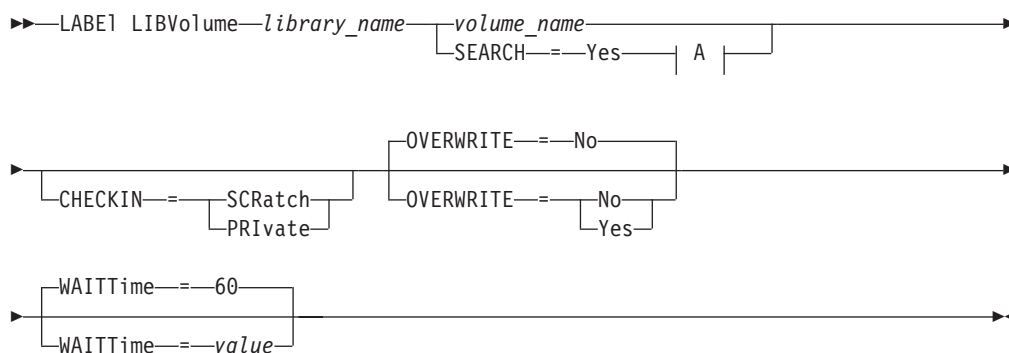
## Syntax for a 349X library



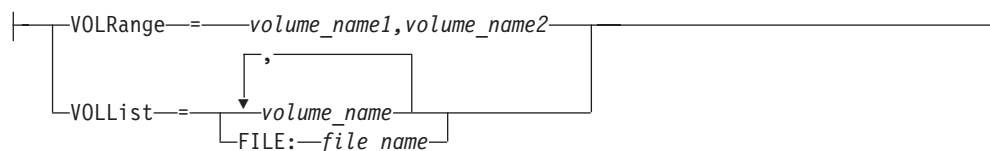
### A (SEARCH=Yes):



## Syntax for an ACSLS library



**A (SEARCH=Yes):**



**Parameters**

***library\_name* (Required)**

Specifies the name of the library that contains the storage volume.

***volume\_name***

Specifies the name of the volume to be labeled.

- For SCSI libraries: The server requests that the volume be inserted into a slot in the library or, if available, into an entry/exit port. The server identifies a slot by the slot's element address. If you are labeling a volume in a SCSI library with multiple entry/exit ports, the volume in the lowest numbered slot will be labeled.
- For MANUAL libraries: The server requests that the volume be inserted into a drive.
- For 349X libraries: The volume might already be in the library, or you might be prompted to put it into the I/O station.

**Remember:** If the specified volume name is already defined in a storage pool or in a volume history file, the volume will not be labeled, and a message will be displayed.

**CHECKIN**

Specifies whether the server checks in the volume. This parameter is optional. The following are possible values:

**SCRatch**

Specifies that the server checks in the volumes and adds them to the library's scratch pool. If a volume has an entry in volume history, you cannot check it in as a scratch volume.

**PRivate**

Specifies that the server checks in the volumes and designates them as private. Private volumes are available only when you request them by name.

If you do not specify a value for this parameter, then the command will only label the volume but will not check it in. If you do not specify a value for this parameter and you want to check in the volume, you must issue the CHECKIN LIBVOLUME command.

**SEARCH**

Specifies that the server searches the library for usable volumes to label. This parameter applies to SCSI, 349X, and ACSLS libraries.

Possible values are:

**Yes**

Specifies that the server labels only volumes that are stored in the library, unless the volume is already labeled or its bar code cannot be read.

## LABEL LIBVOLUME

If you specify the LABELSOURCE=PROMPT option, the volume is moved into the drive from its location in the library or entry and exit ports. The server prompts you to issue the REPLY command containing the label string, and that label is written to the tape.

### Bulk

Specifies that the server searches the library entry/exit ports for usable volumes to label. This option is only valid for SCSI libraries.

If you specify LABELSOURCE=BARCODE, the volume bar code is read, and the tape is moved from its location in the library or in the entry/exit ports to a drive where the bar code label is written. After the tape is labeled, it is moved back to its location in the library, to the entry/exit ports, or to a storage slot if the CHECKIN option is specified. For bar code support to work correctly for libraries supported by Tivoli Storage Manager, the Tivoli Storage Manager server and the device driver must be at the same level. Bar code support is available for libraries supported by Tivoli Storage Manager and which use the Tivoli Storage Manager device driver or the IBM Magstar™ or LTO Ultrium device driver.

**Tip:** You can use the VOLRANGE or VOLLIST parameter to limit the search.

### VOLRange

Specifies a range of volume names separated by a comma. Use this parameter to limit the search for volumes to be labeled when you specify SEARCH=YES (349X, ACSLS, and SCSI libraries) or SEARCH=BULK (SCSI libraries only). If there are no volumes in the library that are within the specified range, the command completes without errors.

You can specify only volume names that can be numerically incremented. In addition to the incremental area, a volume name can include an alphanumeric prefix and an alphanumeric suffix, for example:

| Parameter              | Description  |
|------------------------|--|
| volrange=bar110,bar130 | The 21 volumes are labeled: bar110, bar111, bar112 ,...bar129, bar130. |
| volrange=bar11a,bar13a | The 3 volumes are labeled: bar11a, bar12a, bar13a.                     |
| volrange=123400,123410 | The 11 volumes are labeled: 123400, 123401, ...123409, 123410.         |

### VOLLIST

Specifies a list of volumes. Use this parameter to limit the search for volumes to be labeled when you specify SEARCH=YES (349X, ACSLS, and SCSI libraries) or SEARCH=BULK (SCSI libraries only). If there are no volumes in the library that are in the list, the command completes without errors. The VOLLIST parameter can also be the source of names to be used to label volumes if the LABELSOURCE parameter is set to VOLLIST. If LABELSOURCE=VOLLIST, you must specify the VOLLIST parameter.

Possible values are:

*volume\_name*

Specifies the names of one or more values that are used for the command.  
For example: VOLLIST=TAPE01,TAPE02.

**FILE:***file\_name*

Specifies the name of a file that contains a list of volumes for the

command. In the file, each volume name must be on a separate line. Blank lines and comment lines that begin with an asterisk are ignored. For example, to use volume *TAPE01*, *TAPE02* and *TAPE03*, create a file named *TAPEVOL* that contains these lines:

```
TAPE01
TAPE02
TAPE03
```

You can specify the volumes for the command as follows:  
VOLLIST=FILE:TAPEVOL.

**Remember:** The file name is case-sensitive.

### **LABELSource**

Specifies how or whether the server reads sequential media labels of volumes. This option is only valid for SCSI libraries. Specify this parameter only when SEARCH=YES or SEARCH=BULK.

Possible values are:

#### **Prompt**

The server prompts for volume names as necessary.

#### **Barcode**

The server attempts to read the bar code label. If the attempt fails, the server will not label the volume and will display a message.

**Important:** For bar code support to work properly, the appropriate device drivers must be installed for the libraries.

#### **Vollist**

This option only applies to SCSI libraries. The server attempts to read the specified file or list of files. If the attempt fails, the server will not label the volumes and will display a message.

### **OVERWRITE**

Specifies whether the server attempts to overwrite existing labels. This parameter is optional. The default is NO. Possible values are:

#### **No**

Specifies that the server only labels unlabeled volumes. For StorageTek VolSafe volumes and WORM optical volumes, the value must be NO.

#### **Yes**

Specifies that the server overwrites existing labels *only* if both the existing label and the prompted or bar code label are not already defined in either the server storage pool or volume history list.

### **WAITTime**

Specifies the number of minutes that the server will wait for you to reply or respond to a request. Specify a value in the range 0-9999. If you want to be prompted by the server, specify a wait time greater than zero. The default value is 60 minutes. For example, suppose the server prompts you to insert a tape into the entry/exit port of a library. If you specified a wait time of 60 minutes, the server will issue a request and wait 60 minutes for you to reply. Suppose, on the other hand, you specify a wait time of 0. If you have already inserted a tape, a wait time of zero will cause the operation to continue without prompting. If you have not inserted a tape, a wait time of zero will cause the operation to fail.

## LABEL LIBVOLUME

### Example: Automatically label library volumes

Label tapes in a SCSI library named AUTO automatically as you are checking the volumes in.

```
label libvolume auto checkin=scratch search=yes labelsource=barcode  
overwrite=yes
```

### Example: Label sequential library volumes

Label 3 volumes from bar11a to bar13a in a SCSI library named ABC. When you issue the following command, the 3 volumes are labeled: bar11a, bar12a, bar13a.

```
label libvolume abc checkin=scratch search=yes volrange=bar11a,bar13a  
labelsource=barcode
```

### Related commands

*Table 175. Commands related to LABEL LIBVOLUME*

| Command            | Description  |
|--------------------|--|
| AUDIT LIBRARY      | Ensures that an automated library is in a consistent state.              |
| CANCEL PROCESS     | Cancels a background server process.                                     |
| CHECKIN LIBVOLUME  | Checks a storage volume into an automated library.                       |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library.                     |
| DEFINE LIBRARY     | Defines an automated or manual library.                                  |
| DEFINE VOLUME      | Assigns a volume to be used for storage within a specified storage pool. |
| QUERY LIBRARY      | Displays information about one or more libraries.                        |
| QUERY LIBVOLUME    | Displays information about a library volume.                             |
| QUERY PROCESS      | Displays information about background processes.                         |
| REPLY              | Allows a request to continue processing.                                 |
| UPDATE LIBVOLUME   | Changes the status of a storage volume.                                  |

---

## LOCK commands

Use the LOCK command to prevent users from accessing the server.

The following is a list of LOCK commands for Tivoli Storage Manager:

- “LOCK ADMIN (Lock out an administrator)” on page 530
- “LOCK NODE (Lock out a client node)” on page 531
- “LOCK PROFILE (Lock a profile)” on page 532

## LOCK ADMIN (Lock out an administrator)

Use this command to prevent 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—*admin\_name*—◄◄

### Parameters

*admin\_name* (Required)

Specifies the name of the administrator to be locked out.

### Example: Lock out an administrator

Lock out the administrator CLAUDIA. Issue the command:

```
lock admin claudia
```

### Related commands

Table 176. Commands related to LOCK ADMIN

| Command      | Description   |
|--------------|---|
| QUERY ADMIN  | Displays information about one or more IBM Tivoli Storage Manager administrators. |
| UNLOCK ADMIN | Enables a locked administrator to access IBM Tivoli Storage Manager.              |



## LOCK NODE (Lock out a client node)

Use this command to prevent a client node from accessing the server. A locked client cannot perform any Tivoli Storage Manager operations, even if the operations are scheduled.

### 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—*node\_name*—◄◄

### Parameters

#### *node\_name* (Required)

Specifies the name of the client node to lock out.

### Example: Lock a specific client node

Lock the client node SMITH.

```
lock node smith
```

### Related commands

Table 177. Commands related to LOCK NODE

| Command     | Description   |
|-------------|---|
| QUERY NODE  | Displays partial or complete information about one or more clients.     |
| UNLOCK NODE | Enables a locked user in a specific policy domain to access the server. |

## LOCK PROFILE (Lock a profile)

Use this command on a configuration manager to temporarily lock a profile so that configuration information is not distributed to subscribing managed servers.

You can use this command when you are making multiple updates to your configuration and do not want to distribute this information until the changes are completed.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶ LOCK PROFILE profile_name [ minutes ]

```

### Parameters

#### *profile\_name* (Required)

Specifies the profile to lock. You can use wildcard characters to indicate multiple names.

#### *minutes*

Specifies the time, in minutes, before Tivoli Storage Manager unlocks the configuration profile. Specify an integer from 0 to 10000. The default is 60 minutes. If you specify 0, the configuration profile will not unlock automatically. Use the UNLOCK PROFILE command to unlock the profile before the time period elapses, or to unlock it if you have specified a value of 0. This parameter is optional.

### Example: Lock a profile for a specific amount of time

Lock a profile named DELTA for 30 minutes.

```
lock profile delta 30
```

### Related commands

Table 178. Commands related to LOCK PROFILE

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                       |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                 |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                    |
| QUERY PROFILE          | Displays information about configuration profiles.                 |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |

*Table 178. Commands related to LOCK PROFILE (continued)*

| Command        | Description  |
|----------------|--|
| UNLOCK PROFILE | Enables a locked profile to be distributed to managed servers. |
| UPDATE PROFILE | Changes the description of a profile.                          |

## MACRO (Invoke a macro)

Use this command to invoke a file from the administrative command line that contains one or more Tivoli Storage Manager administrative commands to be performed.

**Restriction:** Use this command with administrative command-line clients only.

A macro is a file that contains one or more Tivoli Storage Manager administrative commands. You can only issue a macro from the administrative client in batch or interactive mode. A macro is stored as a file on the administrative client machine (or system). Macros are not distributed across servers and cannot be scheduled on the server.

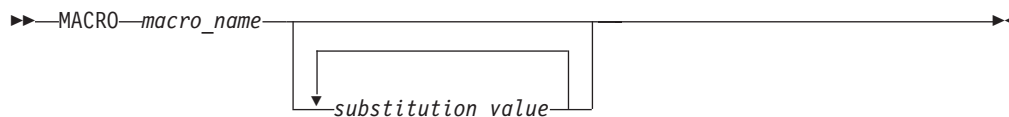
Creating a macro to enter commands can be helpful when you want to issue commands that are used repeatedly, to issue commands that contain several parameters, or to process related commands in a specific order. After you create a macro, you can update the information it contains and use it again, or you can copy the macro file, make changes to the copy, and then run the copy.

For detailed information on macros, how to use them, and the differences between command scripts and administrative command-line client macros, refer to the *Administrator's Guide*.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *macro\_name* (Required)

Specifies the name of the macro.

#### *substitution\_value*

Specifies the value for a substitution variable in a macro. When you use a substitution variable, you can reuse a macro whenever you need to perform the same task for different objects or with different parameter values. To specify a value that contains blanks, you must enclose the value in quotation marks. This parameter is optional.

### Example: Create a macro to register a new administrator

Create a macro file named REGNG. Use the macro to register and grant authority to a new administrator. Write the macro as follows:

```

/* Register and grant authority to a new administrator */
REGister Admin jones passwd          -
CONtactinfo="x1235"                  -
GRant AUTHority jones                 -
Classes=Policy
  
```

Issue the following command to run the macro.

```
macro regng.mac
```

### Example: Write a macro using substitution variables

Create a macro file named AUTHRG, containing substitution variables, to register and grant authority to a new administrator. Write the macro as follows:

```
/* Register and grant authority to 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 */
```

Issue a command similar to the following, entering the values you want to pass to the server to process the command when you run the macro.

```
macro authrg.mac jones passwd x1235 Policy
```

### Related commands

Table 179. Commands related to MACRO

| Command  | Description  |
|----------|--|
| COMMIT   | Makes changes to the database permanent.   |
| ROLLBACK | Discards any uncommitted changes to the database since the last COMMIT was executed. |

## MIGRATE STGPOOL (Migrate storage pool to next storage pool)

Use this command to migrate files from one storage pool to the next storage pool in the storage hierarchy.

This command can only be used with primary storage pools. The storage pool data format cannot be NETAPPDUMP, CELERRADUMP, or NDMPDUMP. Data cannot be migrated into or out of storage pools defined with a CENTERA device class.

Only one migration or reclamation process for a given storage pool is allowed at any given time. If a migration or reclamation process is already running for the storage pool, you cannot start another migration process for the storage pool.

You should only use this command if you are not going to use automatic migration for the storage pool. To prevent automatic migration from running, set the HIGHMIG attribute of the storage pool definition to 100.

The MIGRATE STGPOOL command will honor the values of the following parameters on the DEFINE STGPOOL and UPDATE STGPOOL commands:

- MIGPROCESS
- MIGDELAY
- MIGCONTINUE
- NEXTPOOL
- LOWMIG. However, you can override the value of this parameter on DEFINE STGPOOL and UPDATE STGPOOL by specifying a value for the LOWMIG parameter on the MIGRATE STGPOOL command.

The MIGRATE STGPOOL command will ignore the value of the HIGHMIG parameter of the storage pool definition. Migration will occur regardless of the value of the HIGHMIG parameter.

This command creates one or more migration processes that can be canceled with the CANCEL PROCESS command. The number of processes is limited by the MIGPROCESS attribute of the storage pool definition. To display information on background processes, use the QUERY PROCESS command.

**Remember:** Migrating data from a primary storage pool that is set up for deduplication to another primary storage pool that is also set up for deduplication removes duplicate data.

### Privilege class

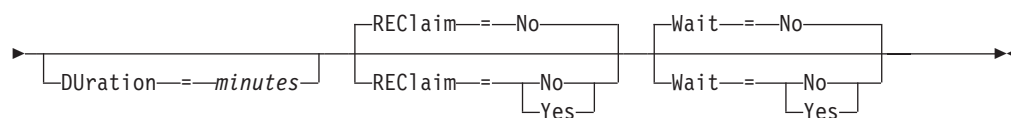
To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for both the storage pool from which the files are to be migrated and the next storage pool to which files are to be migrated.

### Syntax

```

>>MIGrate STGpool—pool_name—┐
                               └─LOWmig—=—number—┘

```



## Parameters

### *pool\_name* (Required)

Specifies the primary storage pool from which files are to be migrated.

### DURATION

Specifies the maximum number of minutes the migration will run before being automatically cancelled. When the specified number of minutes elapses, the server will automatically cancel all migration processes for this storage pool. As soon as the processes recognize the automatic cancellation, they will end. As a result, the migration may run longer than the value you specified for this parameter. You can specify a number from 1 to 9999. This parameter is optional. If not specified, the server will stop only after the low migration threshold is reached.

### LOWmig

For random-access and sequential-access disk storage pools, specifies that the server stops migration when the amount of data in the pool is at or below this percentage of the pool's estimated capacity. The calculation for sequential-access disk storage pools includes the capacity of all the scratch volumes specified for the pool. For other types of sequential-access storage pools, this parameter specifies that the server stops migration when the ratio of volumes containing data to the total number of volumes in the storage pool is at or below this percentage. The total number of volumes includes the maximum number of scratch volumes. This parameter is optional. You can specify a number from 0 to 99. The default value is the LOWMIG attribute of the storage pool definition.

When the storage pool reaches the low migration threshold, the server does not start migration of another node's files. However, because all file spaces that belong to a node or all file spaces for all nodes belonging to a collocation group are migrated together, the occupancy of the storage pool can fall below the value you specified for this parameter. To permit migration to empty the storage pool, you can set LOWMIG=0.

If the storage pool is already below the low migration threshold, no data will be migrated.

### RECLAIM

Specifies whether reclamation is attempted for the storage pool prior to performing the migration. This parameter can only be specified for a sequential-access storage pool. This parameter is optional. The default is No. Possible values are:

#### No

Specifies that the server will not attempt a reclamation prior to starting the migration.

#### Yes

Specifies that the server will attempt reclamation prior to starting the migration. Any volumes in the storage pool that meet the reclamation threshold as specified by the RECLAIM attribute of the storage pool definition will be reclaimed prior to performing the migration. If no volumes meet the reclamation threshold or if, after reclamation, the

## MIGRATE STGPOOL

LOWMIG threshold has not been reached, the server will begin the migration. Before reclaiming space for storage pools defined with RECLAMATIONTYPE=SNAPLOCK, the server will delete all empty WORM FILE volumes during reclamation processing that have exceeded their reclaim period.

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. This default is No. Possible values are:

### No

Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed. Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files may have already been migrated prior to the cancellation.

### Yes

Specifies that the server processes this command in the foreground. The operation must complete before you can continue with other tasks. The server then displays the output messages to the administrative client when the operation completes. Messages are also displayed either in the activity log or the server console, or both, depending on where the messages are logged.

**Note:** You cannot specify WAIT=YES from the server console.

## Example: Migrate a storage pool to the next storage pool

Migrate data from the storage pool named BACKUPPOOL to the next storage pool. Specify that the server should end the migration as soon as possible after 90 minutes.

```
migrate stgpool backuppool duration=90
```

## Related commands

Table 180. Commands related to MIGRATE STGPOOL

| Command         | Description                                    |
|-----------------|--|
| CANCEL PROCESS  | Cancels a background server process.           |
| QUERY PROCESS   | Displays information about background process. |
| QUERY STGPOOL   | Displays information about storage pools.      |
| RECLAIM STGPOOL | Performs reclamation for the storage pool.     |



---

## MOVE commands

Use the MOVE commands to either transfer backup or archive data between storage pools, or to move disaster recovery media on and off site.

The following is a list of MOVE commands for Tivoli Storage Manager:

- “MOVE DATA (Move files on a storage pool volume)” on page 540
- “MOVE DRMEDIA (Move disaster recovery media offsite and back onsite)” on page 544
- “MOVE GRPMEMBER (Move a server group member)” on page 558
- “MOVE MEDIA (Move sequential access storage pool media)” on page 559
- “MOVE NODEDATA (Move data by node in a sequential access storage pool)” on page 567

### MOVE DATA (Move files on a storage pool volume)

Use this command to move files from one storage pool volume to other storage pool volumes.

You can move files from a primary storage pool volume only to volumes in the same or a different primary storage pool. You can move files from a copy storage pool volume only to volumes in the same copy storage pool. You can move files from an active-data pool volume only to volumes in the same active-data pool.

In addition to moving data from volumes in storage pools that have NATIVE or NONBLOCK data formats, this command also lets you move data from volumes in storage pools that have NDMP data formats (NETAPPDUMP, CELERRADUMP, or NDMPDUMP). The target storage pool must have the same data format as the source storage pool. If you are moving data out of a storage pool for the purpose of upgrading to new tape technology, the target primary storage pool must be associated with a library that has the new device for the tape drives. Tivoli Storage Manager supports backend data movement for NDMP images. For details, see the *Administrator's Guide*.

You cannot move data into or out of a storage pool defined with a CENTERA device class.

If you are moving files to volumes in the same storage pool, sufficient space must be available on the volumes. Otherwise, the operation fails.

When you move files from a sequential access volume, multiple sequential access volume mounts are required to move files that span volumes.

When you move files from a random access volume, the server erases any cached copies of files on the volume.

After a move data operation completes, a volume may not be empty if one or more files cannot be relocated to another volume because of input/output errors on the device or because errors were found in the file. If needed, you can delete the volume using the option to discard any data. The files with I/O or other errors are then deleted.

You can use this command to move files from an offsite volume in a copy storage pool or active-data pool. Because the offsite volume cannot be mounted, the server obtains the files that are on the offsite volume from either a primary storage pool, another copy storage pool, or another active-data pool. These files are then written to the destination volumes in the original copy storage pool or active-data pool.

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. If you issue the MOVE DATA command during a restore operation and you receive an error message indicating that one or more files are locked and cannot be moved, you need to reissue the MOVE DATA command after the restore operation has completed in order to move any remaining files.

#### **Remember:**

Issuing this command removes duplicate data when:

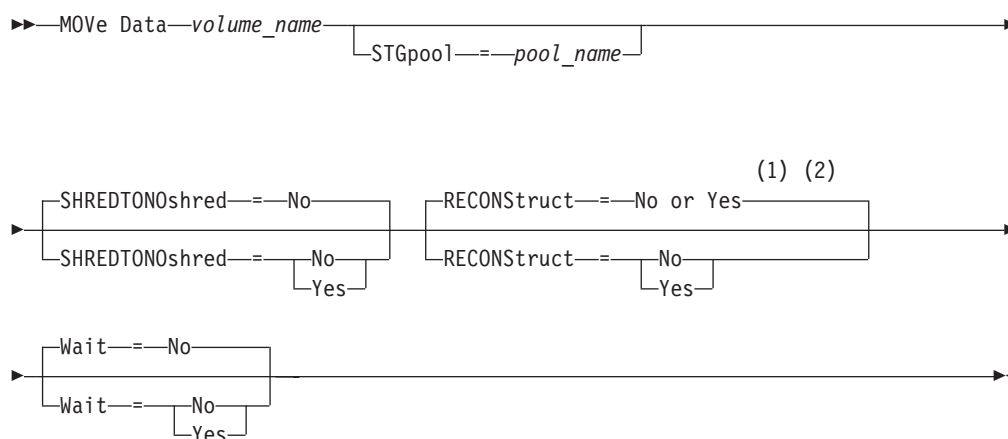
- Moving data from a primary storage pool that is set up for deduplication to another primary storage pool that is also set up for deduplication.

- Moving data within a copy storage pool that is set up for deduplication.
- Moving data within an active-data pool that is set up for deduplication.

## 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



## Notes:

- 1 The default is NO if either the source or target storage pool is random access. The default is YES if both the source and target storage pools are sequential access.
- 2 This parameter is not available or is ignored if the data format is NETAPPDUMP, CELERRADUMP, or NDMPDUMP data.

## Parameters

### *volume\_name* (Required)

Specifies the storage pool volume from which to move files.

### STGpool

Specifies the primary storage pool to which you want to move files (the target storage pool). This parameter is optional and applies only to moving data from primary storage pool volumes. If you do not specify a value for this parameter, files are moved to other volumes within the same storage pool.

### SHREDTONOshred

Specifies whether data will be moved from a storage pool that enforces shredding to a storage pool that does not enforce shredding. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that the server will not allow data to be moved from a storage pool that enforces shredding to a storage pool that does not enforce shredding. If the source storage pool enforces shredding and the target storage pool does not, the operation will fail.

## MOVE DATA

### Yes

Specifies that the server will allow data to be moved from a storage pool that enforces shredding to a storage pool that does not enforce shredding. The source data will be shredded when the operation is complete. The target data will not be shredded when it is deleted.

### RECONStruct

Specifies whether to reconstruct file aggregates during data movement. Reconstruction removes empty space that has accumulated during deletion of logical files from an aggregate. This parameter is optional. If both the source and target storage pools are sequential access, the default value is YES. If either the source or target storage pool is random access, the default is NO.

The parameter is not available or is ignored if the data format is NETAPPDUMP, CELERRADUMP, or NDMPDUMP.

Possible values are:

### No

Specifies that reconstruction of file aggregates is not performed during data movement.

### Yes

Specifies that reconstruction of file aggregates is performed during data movement. You can only specify this option when both the source and the target storage pools are sequential-access.

**Attention:** Reconstruction removes inactive backup files in active-data pools. Specifying RECONSTRUCT=NO when moving data from volumes in an active-data pool will prevent the inactive backup files from being removed.

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default value is No. Possible values are:

### No

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

The server displays messages that are created from the background process either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If a MOVE DATA background process is canceled, some files may have already moved prior to the cancellation.

### Yes

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

### Example: Move files on a storage pool volume

Move files from storage pool volume STGVOL.1 to any available volumes assigned to the 8MMPOOL storage pool.

```
move data stgvol.1 stgpool=8mmpool
```

## Related commands

*Table 181. Commands related to MOVE DATA*

| Command           | Description  |
|-------------------|--|
| CANCEL PROCESS    | Cancels a background server process.                                     |
| DEFINE VOLUME     | Assigns a volume to be used for storage within a specified storage pool. |
| DELETE VOLUME     | Deletes a volume from a storage pool.                                    |
| MOVE DRMEDIA      | Moves DRM media on-site and off-site.                                    |
| QUERY ACTLOG      | Displays messages from the server activity log.                          |
| QUERY DRMEDIA     | Displays information about disaster recovery volumes.                    |
| QUERY PROCESS     | Displays information about background processes.                         |
| QUERY SHREDSTATUS | Displays information about data waiting to be shredded.                  |
| SHRED DATA        | Manually starts the process of shredding deleted data.                   |

## MOVE DRMEDIA (Move disaster recovery media offsite and back onsite)

Use this command to track database backup and copy storage pool volumes that are to be moved offsite and to identify the expired or empty volumes that are to be moved onsite.

The database backup volumes can be for full plus incremental or snapshot backups. You cannot specify virtual volumes (backup objects stored on another server). You can change volumes through each state, or you can use the TOSTATE parameter and skip states to simplify the movements.

**Remember:** The MOVE DRMEDIA command always processes copy storage-pool volumes. (For more information, see the description of the **COPYSTGPOOL** parameter on this command). By default, volumes in active-data pools are not eligible for processing by the disaster recovery manager. To process active-data pool volumes, you must issue the SET DRMACTIVEDATASTGPOOL command, or you must use the ACTIVEDATASTGPOOL parameter on the MOVE DRMEDIA command. To control whether the command processes database backup volumes, you can use the SOURCE parameter on this command.

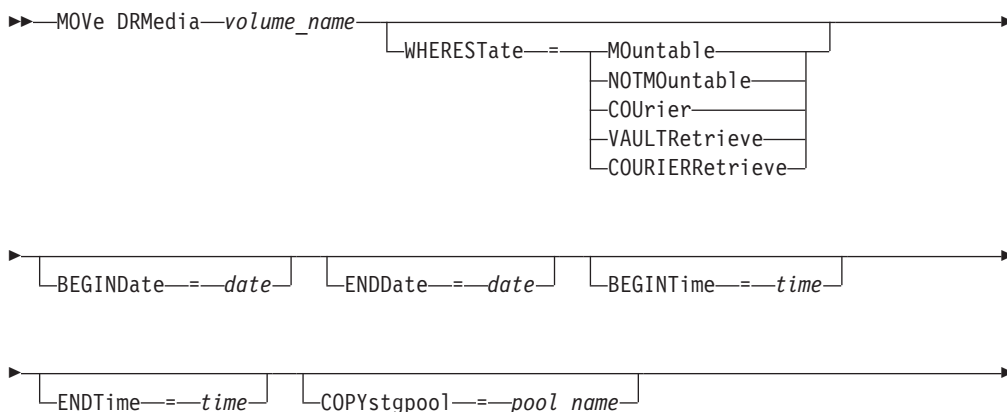
You can use the QUERY ACTLOG command to see if the MOVE DRMEDIA command was successful. You can also view this information from the server console.

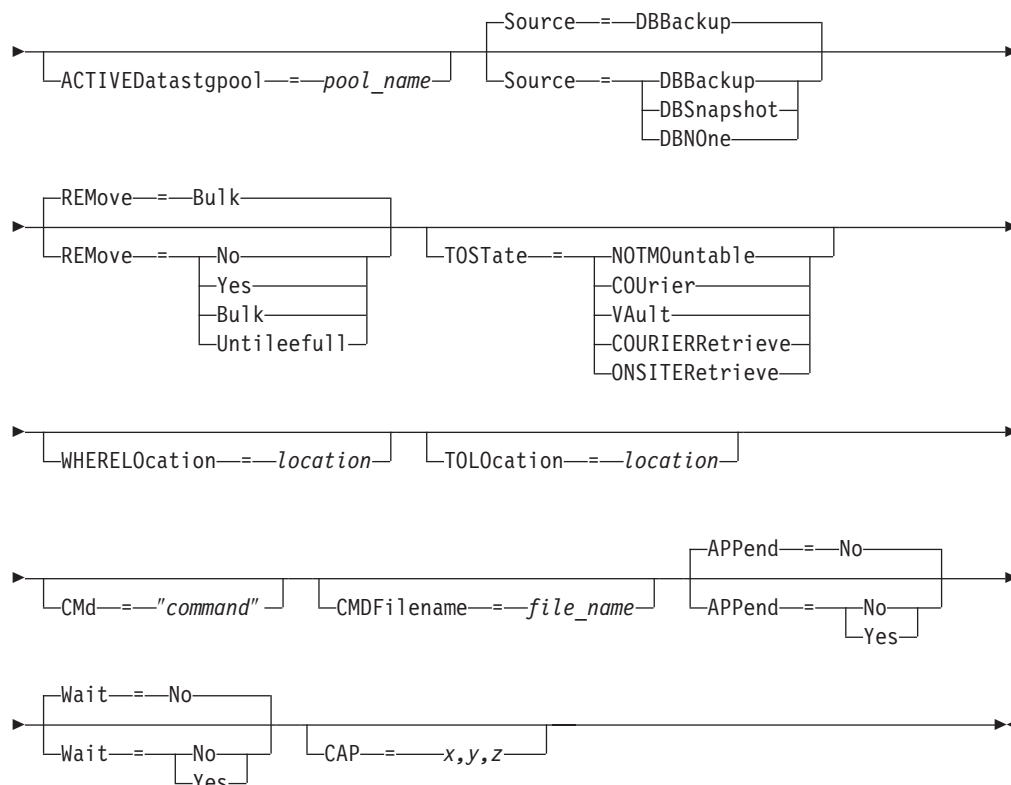
### Privilege class

To issue this command, you must have one of the following privilege classes:

- If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO: operator, unrestricted storage, or system privilege.
- If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES (the default): system privilege.

### Syntax





## Parameters

### *volume\_name* (Required)

Specifies the name of the database backup or copy storage pool volume to be processed. If you use wildcard characters to specify this name, you must also specify WHERESTATE.

### WHERESTATE

Specifies the state of volumes to be processed. This parameter is required if the TOSTATE is not specified or if you use a wildcard character in the volume name. For more information, see Table 186 on page 553 and Table 187 on page 554. Possible values are:

#### MOUNtable

These volumes contain valid data and are available for onsite processing. They will change to NOTMOUNTABLE if the TOSTATE is not specified.

Depending on the behavior of the REMOVE parameter, Tivoli Storage Manager might eject volumes in an automated library before changing the destination state.

For external libraries, the server sends requests to the external library manager to eject the volumes. It depends on the external library manager as to whether or not the volumes are ejected from the library. Refer to the external library documentation for information about the procedures that you should follow when using the MOVE DRMEDIA command to track the volumes.

#### NOTMOUNtable

These volumes are onsite, contain valid data, and are not available for onsite processing. They will change to COURIER if the TOSTATE is not specified.

**COURier**

These volumes are with the courier and being moved offsite. They will change only to VAULT.

**VAULTRetrieve**

These volumes are located at the offsite vault and do not contain valid data. They will change to COURIERRETRIEVE if the TOSTATE is not specified.

**COURIERRetrieve**

These volumes are with the courier and being moved onsite. They will change only to ONSITERETRIEVE. Tivoli Storage Manager deletes from the database the volume records of the database backup and scratch copy storage pool volumes.

**BEGINDate**

Specifies the beginning date used to select volumes. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or after the specified date. The default is the earliest date for which volume information exists.

You can specify the date using one of the following values:

| Value                      | Description                           | Example  |
|----------------------------|---------------------------------------|--|
| <i>MM/DD/YYYY</i>          | A specific date                       | 09/15/1998   |
| TODAY                      | The current date                      | TODAY  |
| <i>TODAY-days or -days</i> | The current date minus days specified | TODAY-7 or -7.<br><br>To identify volumes that were changed to their current state a week ago, you can specify TODAY-7 or simply -7. |

**ENDDate**

Specifies the ending date used to select volumes. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or before the specified date. The default is the current date.

You can specify the date using one of the following values:

| Value                      | Description  | Example  |
|----------------------------|--|--|
| <i>MM/DD/YYYY</i>          | A specific date  | 09/15/1998   |
| TODAY                      | The current date   | TODAY<br><br>To identify volumes that were changed to their current state today, specify TODAY.                                      |
| <i>TODAY-days or -days</i> | The current date minus days specified. The maximum number of days is 9999. | TODAY-1 or -1.<br><br>To identify volumes that were changed to their current state a week ago, you can specify TODAY-1 or simply -1. |

**BEGINTime**

Specifies the beginning time used to select volumes for processing. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA



command has changed the volume to its current state on or after the specified time and date. The default is midnight (00:00:00) on the date specified with the BEGINDATE parameter.

You can specify the time using one of the following values:

| Value                                  | Description  | Example  |
|--|--|--|
| <i>HH:MM:SS</i>                        | A specific time on the specified begin date                          | 12:33:28   |
| NOW                                    | The current time on the specified begin date                         | NOW  |
| NOW+ <i>HH:MM</i><br>or + <i>HH:MM</i> | The current time plus hours and minutes on the specified begin date  | NOW+03:00 or +03:00.   |
| NOW- <i>HH:MM</i><br>or - <i>HH:MM</i> | The current time minus hours and minutes on the specified begin date | NOW-03:30 or -03:30.<br><br>If you issue the MOVE DRMEDIA command at 9:00 with<br>BEGINTIME=NOW-03:30 or<br>BEGINTIME= -03:30, Tivoli Storage Manager identifies the volumes that were changed to their current state at 5:30 on the begin date you specify. |

#### ENDTime

Specifies the ending time used to select volumes for processing. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or after the specified time and date. The default is 23:59:59.

You can specify the time using one of the following values:

| Value                                  | Description  | Example  |
|--|--|--|
| <i>HH:MM:SS</i>                        | A specific time on the specified end date                          | 12:33:28   |
| NOW                                    | The current time on the specified end date                         | NOW  |
| NOW+ <i>HH:MM</i><br>or + <i>HH:MM</i> | The current time plus hours and minutes on the specified end date  | NOW+03:00 or +03:00.<br><br>If you issue the MOVE DRMEDIA command at 9:00 with<br>ENDTIME=NOW+03:30 or<br>ENDTIME=+03:30, Tivoli Storage Manager identifies the volumes that were changed to their current state at 12:30 on the end date you specify. |
| NOW- <i>HH:MM</i><br>or - <i>HH:MM</i> | The current time minus hours and minutes on the specified end date | NOW-03:30 or -03:30.   |

#### COPYstgpool

Specifies the name of the copy storage pool whose volumes are to be processed. This parameter is optional. You can use wildcard characters to specify this name. If you use wildcard characters to specify this name, you must also specify WHERESTATE.

The copy storage pools specified with this parameter override those specified with the SET DRMCOPYSTGPOOL command. If this parameter is not specified, Tivoli Storage Manager selects the storage pools as follows:

- If the SET DRMCOPYSTGPOOL command was previously issued with valid copy storage pool names, Tivoli Storage Manager processes only those storage pools.
- If the SET DRMCOPYSTGPOOL command has not been issued, or if all of the copy storage pools have been removed using the SET DRMCOPYSTGPOOL command, Tivoli Storage Manager processes all copy storage pool volumes in the specified state (MOUNTABLE, NOTMOUNTABLE, COURIER, VAULTRETRIEVE, or COURIERRETRIEVE).

### Source

Specifies whether to include database backup volumes for processing. This parameter is optional. The default is DBBACKUP. The choices are:

#### DBBackup

Specifies that Tivoli Storage Manager includes full and incremental database backup volumes for processing.

#### DBSnapshot

Specifies that Tivoli Storage Manager includes database snapshot backup volumes for processing.

#### DBNone

Specifies that Tivoli Storage Manager does not include any database backup volumes for processing.

### ACTIVEDatastgpool

Specifies the name of the active-data pool whose volumes are to be processed. This parameter is optional. You can use wildcard characters to specify this name. If you use wildcard characters to specify this name, you must also specify the WHERESTATE parameter.

The active-data pools that are specified with this parameter override those that are specified with the SET DRMACTIVEDATASTGPOOL command. If this parameter is not specified, Tivoli Storage Manager selects the storage pools in the following way:

- If the SET ACTIVEDATASTGPOOL command was previously issued with valid active-data pool names, Tivoli Storage Manager processes only those storage pools.
- If the SET ACTIVEDATASTGPOOL command has not been issued, or all of the active-data pools have been removed using the SET ACTIVEDATASTGPOOL command, Tivoli Storage Manager processes all active-data pool volumes in the specified state (NOTMOUNTABLE, COURIER, VAULTRETRIEVE, or COURIERRETRIEVE). Volumes in the MOUNTABLE state are not processed.

**Remember:** You can use the CMDFILE parameter with the MOVE DRMEDIA command to create a file of executable commands to process active-data pool volumes. For example, you can use CMDFILE with the MOVE DRMEDIA \* WHERESTATE=COURIERRETRIEVE TOSTATE=ONSITERETRIEVE command to generate the CHECKIN LIBVOL commands to insert tape volumes into the library.

### REMove

Specifies whether Tivoli Storage Manager ejects the volumes from the library. This parameter is optional and applies only when moving from the

MOUNTABLE state. Possible values are YES, BULK, NO, and UNTILEEFULL. The default is BULK. The response of the server to each option depends on the particular type of library. The tables that follow describe how the server responds for different types of libraries.

Table 182. How the Tivoli Storage Manager server responds for 349X libraries

| REMOVE=YES or REMOVE=NO   | REMOVE=BULK or REMOVE=UNTILEEFULL   |
|---|---|
| The 3494 Library Manager ejects the cartridge to the convenience I/O station. | The 3494 Library Manager ejects the cartridge to the high-capacity output facility. |

Table 183. How the Tivoli Storage Manager server responds for SCSI libraries

| If a library . . .  | And REMOVE=YES, then...  | And REMOVE=BULK, then...  | And REMOVE=NO, then...  | And REMOVE=UNTILEEFULL then...  |
|---|--|---|---|---|
| <i>Does not have entry/exit ports</i>                           | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server then prompts you to remove the cartridge from the slot and to issue a REPLY command. | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a REPLY command. | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a REPLY command. | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server does not prompt you to remove the cartridge and does not require a REPLY command. |
| <i>Has entry/exit ports and an entry/exit port is available</i> | The server moves the cartridge to the available entry/exit port and specifies the port address in a message.<br><br>The server then prompts you to remove the cartridge from the slot and to issue a REPLY command.        | The server moves the cartridge to the available entry/exit port and specifies the port address in a message.<br><br>The server does not prompt you to remove the cartridge and does not request a REPLY command.        | The server specifies the port address in a message.<br><br>The server does not prompt you to remove the cartridge and does not request a REPLY command.   | The server moves the cartridge to the available entry/exit port and specifies the port address in a message.<br><br>The server does not prompt you to remove the cartridge and does not request a REPLY command.        |
| <i>Has entry/exit ports, but no ports are available</i>         | The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.<br><br>The server then prompts you to remove the cartridge from the slot and to issue a REPLY command. | The server waits for a port to be made available.   | The server specifies the port address in a message.<br><br>The server does not prompt you to remove the cartridge and does not request a REPLY command.   | The command fails and any remaining eligible volumes are not processed.<br><br>Make the port available and issue the command again.   |

Table 184. How the Tivoli Storage Manager server responds for ACSLS libraries

| <b>REMOVE=YES, REMOVE=BULK, or REMOVE=UNTILEEFULL</b>   | <b>REMOVE=NO</b>  |
|---|---|
| The server ejects the cartridge to the convenience I/O station.   | The server does not eject the cartridge.  |
| The server then deletes the volume entry from the server library inventory.   | The server deletes the volume entry from the server library inventory and leaves the volume in the library. |
| While moving volumes from the MOUNTABLE state with REMOVE=YES specified, the MOVE MEDIA command uses more than one slot in the CAP for a StorageTek library with ACSLS. |   |

Table 185. How the Tivoli Storage Manager server responds for external libraries

| <b>REMOVE=YES, REMOVE=BULK, REMOVE=UNTILEEFULL, or REMOVE=NO</b>  |
|---|
| The server requests the external library manager to eject the volume from the library.  |
| It depends on the external library manager as to whether or not the volume is ejected from the library. Refer to the external library documentation for information about the procedures that you should follow when using the MOVE DRMEDIA command to track volumes. |

**TOSTate**

Specifies the destination state of the volumes being processed. This parameter is required if WHERESTATE is not specified. If you specify TOSTATE but not WHERESTATE, you must specify the volume name. Wildcard characters are not allowed. For more information, see Table 186 on page 553 and Table 187 on page 554.

Possible values are:

**NOTMOUNTable**

Specifies that volumes are to change to the NOTMOUNTABLE state. This value is valid only if the volumes are in the MOUNTABLE state.

If volumes are in an automated library, Tivoli Storage Manager might eject, depending on the behavior of the REMOVE parameter, the volumes from the library before changing them to the NOTMOUNTABLE state.

For external libraries, the server sends requests to the external library manager to eject the volumes. It depends on the external library manager as to whether or not the volumes are ejected from the library. Refer to the external library documentation for information about the procedures that you should follow when using the MOVE DRMEDIA command to track the volumes.

**COURier**

Specifies that volumes are to change to the COURIER state. This value is valid only if the volumes are in the MOUNTABLE or NOTMOUNTABLE state.

Depending on the behavior of the REMOVE parameter and whether volumes are in an automated library, Tivoli Storage Manager might eject the volumes from the library before changing them to the COURIER state.

For external libraries, the server sends requests to the external library manager to eject the volumes. It depends on the external library manager as to whether or not the volumes are ejected from the library. Refer to the

external library documentation for information about the procedures that you should follow when using the MOVE DRMEDIA command to track the volumes.

### **VAult**

Specifies that volumes are to change to the VAULT state. This value is valid only if the volumes are in the MOUNTABLE, NOTMOUNTABLE, or COURIER state.

Depending on the behavior of the REMOVE parameter and whether volumes are in an automated library, Tivoli Storage Manager might eject the volumes from the library before changing them to the VAULT state.

For external libraries, the server sends requests to the external library manager to eject the volumes. It depends on the external library manager as to whether or not the volumes are ejected from the library. Refer to the external library documentation for information about the procedures that you should follow when using the MOVE DRMEDIA command to track the volumes.

### **COURIERRetrieve**

Specifies that volumes are to change to the COURIERRETRIEVE state. This value is valid only if the volumes are in the VAULTRETRIEVE state.

### **ONSITERetrieve**

Specifies that volumes are to change to the ONSITERETRIEVE state. This value is valid only if the volumes are in the VAULTRETRIEVE or COURIERRETRIEVE state. For database backup and scratch copy storage pool volumes that are changing to the ONSITERETRIEVE state, Tivoli Storage Manager deletes the volume records from the database.

### **WHERELocation**

Specifies the current location of the volumes. This parameter is optional. The maximum length of the location is 255 characters. Enclose the text in quotation marks if it contains any blank characters.

### **TOLocation**

Specifies the destination location of the volumes. This parameter is optional. The maximum length of the location specified is 255 characters. Enclose the text in quotation marks if it contains any blank characters. If you do not specify the destination location, the location defined by the SET DRMNOTMOUNTABLE command is used.

### **CMd**

Specifies the creation of executable commands to process each volume obtained by this command. This parameter is optional. The maximum length of this command specification is 255 characters. DRM writes the commands to a file that is specified by the CMDFILENAME parameter. If the length of the written command is greater than 240 characters, it is split into multiple lines and continuation characters (+) are added. You may need to alter the continuation character according to the product that runs the commands.

#### *command*

The command string enclosed in quotation marks. The string must not include embedded quotation marks. For example, this is a valid CMD parameter:

```
cmd="checkin libvol lib8mm &vol status=scratch"
```

This is an example of a CMD parameter that is *not* valid:

```
cmd=""checkin libvol lib8mm" &vol status=scratch""
```

The command can include substitution variables. The variables are not case-sensitive, and must not contain blank spaces after the ampersand (&). The possible variables are:

**&VOL**

A volume name.

**&LOC**

A volume location.

**&VOLDNS**

The file name to be written into the sequential access media labels. For example, if the applicable device class sets TSM as the tape volume prefix, a copy storage pool tape volume file name might be TSM.BFS and a database backup tape volume file name might be TSM.DBB.

**&NL**

The new line character. When &NL is specified, the command is split at the &NL variable. If required, you must specify the appropriate continuation character before the &NL. If the &NL is not specified and the command line is greater than 240 characters, the line is split into multiple lines and continuation characters (+) are added.

**CMDFilename**

Specifies the fully qualified name of the file that will contain the commands specified by CMD parameter. This parameter is optional.

If you do not specify a file name or if you specify a null string (""), DRM uses the file name specified by the SET DRMCMDFILENAME. If you do not specify a file name with the SET DRMCMDFILENAME command, DRM generates a file name by appending exec.cmds to the directory path name of the current working directory of the Tivoli Storage Manager server.

If the operation fails after the command file has been created, the file is not deleted.

**APPend**

Specifies whether to overwrite any existing contents of the command file or append the commands to the file. This parameter is optional. The default is NO. Possible values are:

**No**

DRM overwrites the contents of the file.

**Yes**

DRM appends the commands to the file.

**Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default value is NO. Possible values are:

**No**

Specifies that Tivoli Storage Manager processes this command in the background.

Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To see if the operation was successful, issue the QUERY ACTLOG command.

**Yes**

Specifies that the server processes this command in the foreground. Wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client.

**Restriction:** You cannot specify WAIT=YES from the server console.

**CAP**

Specifies which cartridge access port (CAP) to use for ejecting volumes if you specify REMOVE=YES. This parameter applies to volumes in ACSLS libraries only. If a CAP priority greater than zero does not exist in the library, this parameter is required. If a CAP priority greater than zero does exist in the library, this parameter is optional. If you do not specify the parameter, the ACSLS server will choose any available CAP with a priority greater than zero.

To display valid CAP identifiers ( $x,y,z$ ), issue the QUERY CAP command with ALL specified from the Automated Cartridge System System Administrator (ACSSA) console on the ACSLS server host. The identifiers are as follows:

- $x$  The Automated Cartridge System (ACS) ID. This identifier can be a number between 0-126.
- $y$  The Library Storage Module (LSM) ID. This identifier can be a number between 0-23.
- $z$  The CAP ID. This identifier can be a number between 0-11.

For more information, refer to your StorageTek documentation.

**Rules for destination states and destination locations**

The following table shows how DRM determines the destination state and location of a volume.

**Destination state**

- TOSTATE specified.
- The next state of the WHERESTATE specified, if the TOSTATE is not specified.

**Destination location**

- TOLOCATION specified.
- The location of the TOSTATE specified, if the TOLOCATION is not specified.
- The location of the next state of the WHERESTATE specified, if the TOLOCATION and TOSTATE are not specified.

*Table 186. Volume destination and location*

| Parameters specified          | Destination state                | Destination location       |
|-------------------------------|----------------------------------|----------------------------|
| WHERESTATE                    | The next state of the WHERESTATE | Location of the next state |
| WHERESTATE TOSTATE            | TOSTATE                          | Location of the TOSTATE    |
| WHERESTATE TOLOCATION         | The next state of the WHERESTATE | TOLOCATON                  |
| WHERESTATE TOSTATE TOLOCATION | TOSTATE                          | TOLOCATION                 |
| TOSTATE                       | TOSTATE                          | Location of the TOSTATE    |



Table 186. Volume destination and location (continued)

| Parameters specified                   | Destination state | Destination location    |
|--|-------------------|-------------------------|
| TOSTATE<br>WHERELOCATION               | TOSTATE           | Location of the TOSTATE |
| TOSTATE<br>WHERELOCATION<br>TOLOCATION | TOSTATE           | TOLOCATION              |

## Rules for state transitions

The following tables show the state transitions that volumes are eligible for, based on their current state.

Table 187. State transitions for volumes

| Volume's current state | Destination state |              |         |       |
|------------------------|-------------------|--------------|---------|-------|
|                        | MOUNTABLE         | NOTMOUNTABLE | COURIER | VAULT |
| MOUNTABLE              | N                 | Y            | Y       | Y     |
| NOTMOUNTABLE           | N                 | N            | Y       | Y     |
| COURIER                | N                 | N            | N       | Y     |
| VAULT                  | N                 | N            | N       | N     |
| VAULTRETRIEVE          | N                 | N            | N       | N     |
| COURIERRETRIEVE        | N                 | N            | N       | N     |
| ONSITERETRIEVE         | N                 | N            | N       | N     |

| Volume's current state | Destination state |                  |                 |
|------------------------|-------------------|------------------|-----------------|
|                        | VAULT-RETRIEVE    | COURIER-RETRIEVE | ONSITE-RETRIEVE |
| MOUNTABLE              | N                 | N                | N               |
| NOTMOUNTABLE           | N                 | N                | N               |
| COURIER                | N                 | N                | N               |
| VAULT                  | N                 | N                | N               |
| VAULTRETRIEVE          | N                 | Y                | Y               |
| COURIERRETRIEVE        | N                 | N                | Y               |
| ONSITERETRIEVE         | N                 | N                | N               |

### Example: Move disaster recovery media from the NOTMOUNTABLE state

Move disaster recovery media that is in the NOTMOUNTABLE state to the COURIER state and then query the results.

```
move drmedia * wherestate=notmountable
tostate=courier
```

```
query actlog search="MOVE DRMEDIA"
```



```

08/11/1999 11:12:24 ANR0984I Process 10 for MOVE DRMEDIA started
                      in the BACKGROUND at 11:12:24.
08/11/1999 11:12:24 ANR0610I MOVE DRMEDIA started by HSIAO as
                      process 10.
08/11/1999 11:12:25 ANR6683I MOVE DRMEDIA: Volume TAPE0P was moved
                      from NOTMOUNTABLE state to COURIER.
08/11/1999 11:12:25 ANR6683I MOVE DRMEDIA: Volume TAPE1P was moved
                      from NOTMOUNTABLE state to COURIER.
08/11/1999 11:12:25 ANR6683I MOVE DRMEDIA: Volume DBTP02 was moved
                      from NOTMOUNTABLE state to COURIER.
08/11/1999 11:12:25 ANR6683I MOVE DRMEDIA: Volume DBTP01 was moved
                      from NOTMOUNTABLE state to COURIER.
08/11/1999 11:12:25 ANR6682I MOVE DRMEDIA command ended: 4 volumes
                      processed.
08/11/1999 11:12:25 ANR0611I MOVE DRMEDIA started by HSIAO as
                      process 10 has ended.
08/11/1999 11:12:25 ANR0985I Process 10 for MOVE DRMEDIA running in
                      the BACKGROUND processed 4 items with a
                      completion state of SUCCESS at 11:12:25.

```

### Example: Move disaster recovery media from the MOUNTABLE state

Move disaster recovery media from the MOUNTABLE state to the COURIER state. If the media is in an automated library, MOVE DRMEDIA ejects the media before changing the state.

```
move drmedia * wherestate=mountable tostate=courier wait=yes
```

```

ANR0984I Process 12 for MOVE DRMEDIA started
in the FOREGROUND at 09:57:17.
ANR0609I MOVE DRMEDIA started as process 12.
ANR0610I MOVE DRMEDIA started by HSIAO as
process 12.
ANR6696I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume TAPE01 in library LIB8MM starting.
ANR6697I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume TAPE01 in library LIB8MM completed
successful.
ANR6683I MOVE DRMEDIA: Volume TAPE01 was moved
from MOUNTABLE state to COURIER.
ANR6696I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume TAPE02 in library LIB8MM starting.
ANR6697I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume TAPE02 in library LIB8MM completed
successful.
ANR6683I MOVE DRMEDIA: Volume TAPE02 was moved
from MOUNTABLE state to COURIER.
ANR6696I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume DBTP05 in library LIB8MM starting.
ANR6697I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume DBTP05 in library LIB8MM completed
successful.
ANR6683I MOVE DRMEDIA: Volume DBTP05 was moved
from MOUNTABLE state to COURIER.
ANR6696I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume DBTP04 in library LIB8MM starting.
ANR6697I MOVE DRMEDIA: CHECKOUT LIBVOLUME for
volume DBTP04 in library LIB8MM completed
successful.
ANR6683I MOVE DRMEDIA: Volume DBTP04 was moved
from MOUNTABLE state to COURIER.
ANR6682I MOVE DRMEDIA command ended: 4 volumes
processed.
ANR0611I MOVE DRMEDIA started by HSIAO as
process 12 has ended.
ANR0985I Process 12 for MOVE DRMEDIA running
in the FOREGROUND processed 4 items with a
completion state of SUCCESS at 10:12:25.

```

## Example: Move disaster recovery media from the VALUTRETRIEVE state

Move disaster recovery media that is in the VALUTRETRIEVE state to the ONSITERETRIEVE state. Generate a CHECKIN LIBVOLUME command for each volume that is successfully processed and store the commands in the file:

```
/drm/move/exec.cmds
move drmedia * wherestate=vaultretrieve tostate=onsiteretrieve
cmdfilename=/drm/move/exec.cmds
cmd="checkin libvol lib8mm &vol status=scratch"
```

Query the results:

```
query actlog search="MOVE DRMEDIA"
```

```
08/13/1999 09:12:24 ANR0984I Process 15 for MOVE DRMEDIA started in
                    the BACKGROUND at 09:12:24.
08/13/1999 09:12:24 ANR0610I MOVE DRMEDIA started by HSIA0 as
                    process 15.
08/13/1999 09:12:24 ANR6684I MOVE DRMEDIA: Volume CSTEP01 was deleted.
08/13/1999 09:12:24 ANR6684I MOVE DRMEDIA: Volume CSTEP02 was deleted.
08/13/1999 09:12:24 ANR6684I MOVE DRMEDIA: Volume DBTP10 was deleted.
08/13/1999 09:12:24 ANR6684I MOVE DRMEDIA: Volume DBTP11 was deleted.
08/13/1999 09:12:27 ANR6682I MOVE DRMEDIA command ended: 4 volumes
                    processed.
08/13/1999 09:12:42 ANR0611I MOVE DRMEDIA started by HSIA0 as process
                    15 has ended.
08/13/1997 09:12:42 ANR0985I Process 15 for MOVE DRMEDIA running in
                    the BACKGROUND processed 4 items with a
                    completion state of SUCCESS at 09:12:42.
```

The volume check-in commands were also created in the file:

```
/drm/move/exec.cmds
```

The file contains these lines:

```
checkin libvol lib8mm CSTEP01 status=scratch
checkin libvol lib8mm CSTEP02 status=scratch
checkin libvol lib8mm DBTP10 status=scratch
checkin libvol lib8mm DBTP11 status=scratch
```

**Remember:** To process the CHECKIN LIBVOLUME commands, issue the MACRO command with the file name as the macro name.

## Related commands

Table 188. Commands related to MOVE DRMEDIA

| Command            | Description  |
|--------------------|--|
| BACKUP DB          | Backs up the IBM Tivoli Storage Manager database to sequential access volumes. |
| BACKUP STGPOOL     | Backs up a primary storage pool to a copy storage pool.                        |
| CANCEL PROCESS     | Cancels a background server process.   |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library.                           |
| DISMOUNT VOLUME    | Dismounts a sequential, removable volume by the volume name.                   |
| PREPARE            | Creates a recovery plan file.  |

Table 188. Commands related to MOVE DRMEDIA (continued)

| Command                   | Description  |
|---------------------------|--|
| QUERY ACTLOG              | Displays messages from the server activity log.  |
| QUERY DRMSTATUS           | Displays DRM system parameters.  |
| QUERY PROCESS             | Displays information about background processes.   |
| QUERY DRMEDIA             | Displays information about disaster recovery volumes.  |
| SET DRMACTIVEDATASTGPOOL  | Specifies that active-data storage pools are managed by DRM.   |
| SET DRMCOPYSTGPOOL        | Specifies that copy storage pools are managed by DRM.  |
| SET DRMCOURIERNAME        | Specifies the name of the courier for the disaster recovery media.   |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.  |
| SET DRMVAULTNAME          | Specifies the name of the vault where DRM media is stored.   |
| SET DRMCMDFILENAME        | Specifies a file name for containing DRM executable commands.  |
| SET DRMFILEPROCESS        | Specifies whether the MOVE DRMEDIA or QUERY DRMEDIA command processes files associated with a device type of file. |
| SET DRMNOTMOUNTABLENAME   | Specifies the location name of the DRM media to be sent offsite.   |

## MOVE GRPMEMBER (Move a server group member)

Use this command to move a member from one server group to another server group. The command fails if the member you are moving has the same name as a current member of the group.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► **MOVE GRPMEMBER** *member\_name* *from\_group* *to\_group* ◄◄

### Parameters

*member\_name* **(Required)**

Specifies the member (a server or a server group) to move.

*from\_group* **(Required)**

Specifies the server group with which the member is currently associated.

*to\_group* **(Required)**

Specifies the new server group for the member.

### Example: Move a server to another server group

Move member PAYSON from REGION1 group to REGION2 group.

```
move grpmember payson region1 region2
```

### Related commands

Table 189. Commands related to MOVE GRPMEMBER

| Command            | Description                                     |
|--------------------|---|
| DEFINE GRPMEMBER   | Defines a server as a member of a server group. |
| DEFINE SERVERGROUP | Defines a new server group.                     |
| DELETE GRPMEMBER   | Deletes a server from a server group.           |
| DELETE SERVERGROUP | Deletes a server group.                         |
| QUERY SERVER       | Displays information about servers.             |
| QUERY SERVERGROUP  | Displays information about server groups.       |
| RENAME SERVERGROUP | Renames a server group.                         |
| UPDATE SERVERGROUP | Updates a server group.                         |

## MOVE MEDIA (Move sequential access storage pool media)

Use this command to manage overflow storage pools. The database tracks media that has been moved using this command.

This command applies to sequential access primary and copy storage pool volumes that are managed by an automated library (including an external library). The library does not have to be full. One or more sequential access storage pool volumes can be processed at the same time.

Use the DAYS parameter to identify eligible volumes to be moved. Use the OVERFLOW LOCATION parameter to record the storage location for the moved media.

This command generates a background process that you can view using the QUERY PROCESS command. To cancel, issue the CANCEL PROCESS command.

To determine if the command was successful, issue the QUERY ACTLOG command or use the server console.

The volumes moved by the MOVE DRMEDIA command for offsite recovery are not processed by the MOVE MEDIA command.

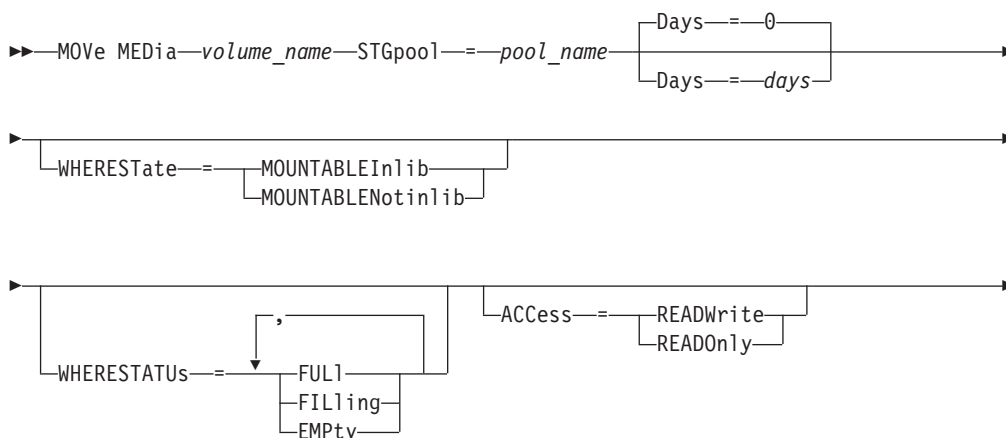
For additional information see the *Administrator's Guide*.

### Privilege class

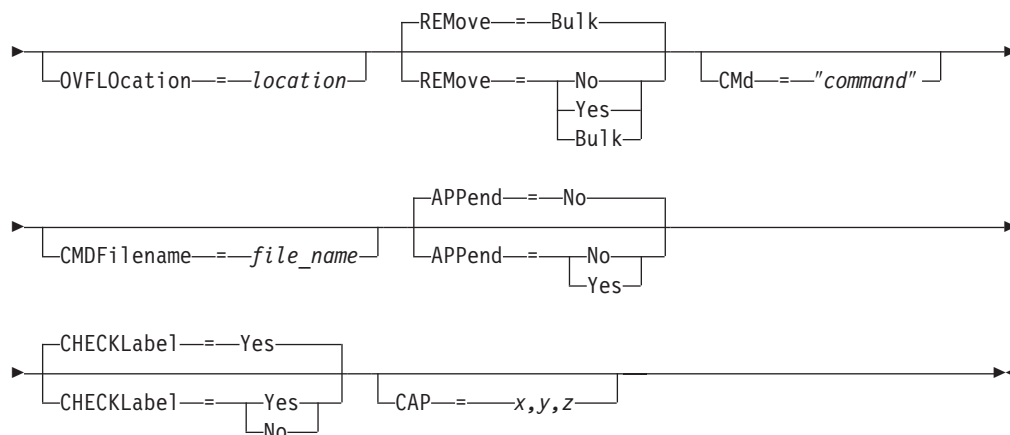
To issue this command, you must have one of the following privilege classes:

- If the CMD parameter is NOT specified: Operator or system privilege.
- If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO: Operator, unrestricted storage, or system privilege.
- If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES (the default): System privilege.

### Syntax



## MOVE MEDIA



### Parameters

#### *volume\_name* (Required)

Specifies the name of the sequential access primary or copy storage pool volume to be processed. You can use a wildcard character to specify the name. All matching volumes are considered for processing.

#### STGpool (Required)

Specifies the name of the sequential access primary or copy storage pool that is used to select the volumes for processing. You can use a wildcard character to specify the name. All matching storage pools are processed. If the storage pool specified is not managed by an automated library, no volumes are processed.

#### Days

Specifies the number of days that must elapse after the volume has been written or read, before the volume is eligible for processing by the command. This parameter is optional. You can specify a number from 0 to 9999. The default value is 0. The most recent of the volumes' last written date or last read date is used to calculate the number of days elapsed.

#### WHEREState

Specifies the current state of the volumes to be processed. This parameter is used to restrict processing to the volumes that are in the specified state. This parameter is optional. The default value is MOUNTABLEINLIB.

Possible values are:

##### **MOUNTABLEInlib**

Specifies that storage pool volumes are to transition from the MOUNTABLEINLIB state to the MOUNTABLENOTINLIB state. Volumes in the MOUNTABLEINLIB state contain valid data and are in the library.

##### **MOUNTABLENotinlib**

Specifies that storage pool volumes are to change from the MOUNTABLENOTINLIB state back to the MOUNTABLEINLIB state. Volumes in the MOUNTABLENOTINLIB state may contain valid data and are in the overflow location.

- For *empty scratch volumes*, the MOVE MEDIA command deletes the volume records so that they can be used again.
- For *private volumes*, the MOVE MEDIA command resets the volume location to blank, changes the volumes' state to CHECKIN, and changes the last update date to the current date.

- For *scratch volumes with data*, the MOVE MEDIA command resets the volume location to blank, changes the volumes' state to CHECKIN, and changes the last update date to the current date.

**Attention:** Volumes in the CHECKIN state may contain valid data and need to be checked into the library.

### WHERESTATUS

Specifies that the move process should be restricted by volume status. This parameter is optional. You can specify more than one status in a list by separating each status with a comma and no intervening spaces. If you do not specify this parameter, volumes moved from the MOUNTABLEINLIB state to the MOUNTABLENOTINLIB state are restricted to only full volumes, and volumes moved from the MOUNTABLENOTINLIB state to the MOUNTABLEINLIB state are restricted to only empty volumes.

Possible values are:

#### FULL

Moves volumes with a status of FULL.

#### FILLing

Moves volumes with a status of FILLING.

#### EMPTy

Moves volumes with a status of EMPTY.

### ACCess

Specifies how users and system processes access files in the storage pool volume that is moved out from an automated library and stored in an overflow location by the MOVE MEDIA command. This parameter is optional. If you do not specify this parameter, moving volumes from the MOUNTABLEINLIB state to the MOUNTABLENOTINLIB process updates the volumes' access mode to READONLY, and moving volumes from the MOUNTABLENOTINLIB state to the MOUNTABLEINLIB process updates the volumes' access mode to READWRITE.

Possible values are:

#### READWrite

Specifies that users and system processes can read from and write to files stored on the volume that is in the overflow location. If this value is specified, Tivoli Storage Manager requests the volume to be checked into the library when the volume is needed for a read or write operation.

#### READOnly

Specifies that users and system processes can only read files that are stored on the volume that is in the overflow location. The server requests the volume to be checked into the library only when the volume is needed for a read operation.

### OVFLocation

Specifies the overflow location that is the destination of the volumes being processed. The maximum length of the location name is 255 characters. The location name information must be enclosed in quotation marks if it contains any blank characters. If you do not specify an overflow location and the storage pool also has no overflow location identified, the server changes the location of the ejected volume to a null string ("").

### REMove

Specifies whether Tivoli Storage Manager ejects the volumes from the library

## MOVE MEDIA

when moving volumes from the MOUNTABLEINLIB state. This parameter is optional. Possible values are YES, BULK, and NO. The default is YES. The response of the server to each option depends on the type of library.

**349X Libraries:** The following table shows how the server responds for 349X libraries.

*Table 190. How the Tivoli Storage Manager Server Responds for 349X Libraries*

| REMOVE=YES or REMOVE=NO   | REMOVE=BULK   |
|---|---|
| The 3494 Library Manager ejects the cartridge to the convenience I/O station. The server then deletes the volume entry from the server library inventory. This option applies to all automated libraries. | The 3494 Library Manager ejects the cartridge to the high-capacity output facility. |

**SCSI libraries:** The following table shows how the server responds to YES, BULK, and NO for SCSI libraries.

*Table 191. How the Tivoli Storage Manager Server Responds for SCSI Libraries*

| If a library . . .  | And REMOVE=YES. . .  | And REMOVE=BULK...   | And REMOVE=NO  |
|---|--|--|--|
| <i>Does not have entry/exit ports</i>                           | <p>The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.</p> <p>The server then prompts you to remove the cartridge from the slot and issue a REPLY command.</p> | <p>The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.</p> <p>The server does not prompt you to remove the cartridge and does not require a REPLY command.</p> | <p>The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.</p> <p>The server does not prompt you to remove the cartridge and does not require a REPLY command.</p> |
| <i>Has entry/exit ports and an entry/exit port is available</i> | <p>The server moves the cartridge to the available entry/exit port and specifies the port address in a message.</p> <p>The server then prompts you to remove the cartridge from the slot and issue a REPLY command.</p>        | <p>The server moves the cartridge to the available entry/exit port and specifies the port address in a message.</p> <p>The server does not prompt you to remove the cartridge and does not request a REPLY command.</p>        | <p>The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.</p> <p>The server does not prompt you to remove the cartridge and does not require a REPLY command.</p> |
| <i>Has entry/exit ports, but no ports are available</i>         | <p>The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.</p> <p>The server then prompts you to remove the cartridge from the slot and issue a REPLY command.</p> | <p>The server waits for an entry/exit port to be made available.</p>   | <p>The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.</p> <p>The server does not prompt you to remove the cartridge and does not require a REPLY command.</p> |

**ACSLs libraries:** The following table shows how the server responds for ACSLS libraries.



Table 192. How the Tivoli Storage Manager Server Responds for ACSLS Libraries

| REMOVE=YES or REMOVE=BULK   | REMOVE=NO   |
|---|---|
| The server ejects the cartridge to the convenience I/O station.   | The server does not eject the cartridge.  |
| The server then deletes the volume entry from the server library inventory.   | The server deletes the volume entry from the server library inventory and leaves the volume in the library. |
| While moving volumes from the MOUNTABLE state with REMOVE=YES specified, the MOVE MEDIA command uses more than one slot in the CAP for a StorageTek library with ACSLS. |   |

**External libraries:** The following table shows how the server responds for external libraries.

Table 193. How the Tivoli Storage Manager Server Responds for External Libraries

| REMOVE=YES or REMOVE=BULK   | REMOVE=NO   |
|---|---|
| The server ejects the cartridge to the convenience I/O station. The server then deletes the volume entry from the server library inventory. | The server does not eject the cartridge.  |
|   | The server deletes the volume entry from the server library inventory and leaves the volume in the library. |

### CMD

Specifies the creation of executable commands. This parameter is optional. You must enclose your command specification in quotation marks. The maximum length of the command specification is 255 characters. For each volume successfully processed by the MOVE MEDIA command, Tivoli Storage Manager writes the associated commands to a file. Specify the file name with the CMDFILENAME parameter.

If you do not specify the file name, the MOVE MEDIA command will generate a default filename by appending the string "exec.cmds.media" to the Tivoli Storage Manager server directory.

If the length of the command written to the file exceeds 255 characters, it is split into multiple lines and a continuation character, +, is added to all but the last line of the command. You may need to alter the continuation character according to the requirements of the product that runs the commands.

If you do not specify CMD, the MOVE MEDIA command will not generate any executable commands.

#### string

Specifies the string to build an executable command. You can specify any free form text for the string. Enclose the full string in quotation marks. For example, the following is a valid executable command specification:

```
CMD="UPDATE VOLUME &VOL"
```

The following is an invalid executable command specification:

```
CMD=""UPDATE VOLUME" &VOL"
```

#### substitution

Specifies a variable for which you want the command to substitute a value. The possible substitution variables are:

#### &VOL

Substitute the volume name for &VOL. You can specify lowercase

characters, &vol. No spaces or blanks are allowed between ampersand, &, and VOL. If there are spaces or blanks between ampersand and VOL, the MOVE MEDIA command will treat them as strings and no substitution will be set. If &VOL is not specified, no volume name is set in the executable command.

### **&LOC**

Substitute the volume location for &LOC. You can specify lowercase characters, &loc. No spaces or blanks are allowed between ampersand, &, and LOC. If there are spaces or blanks between ampersand and LOC, the MOVE MEDIA command will treat them as strings and no substitution will be set. If &LOC is not specified, no location name is set in the executable command.

### **&VOLDSN**

Substitute the volume file name for &VOLDSN. An example of a storage pool tape volume file name using the default prefix ADSM is ADSM.BFS. If &VOLDSN is not specified, no volume file name is set in the executable command.

### **&NL**

Substitute a new line character for &NL. When &NL is specified, the MOVE MEDIA command splits the command at the position where the &NL is and will not append any continuation character. The user is responsible for specifying the proper continuation character before the &NL if one is required. The user is also responsible for the length of the line written. If the &NL is not specified and the length of the command line exceeds 255, the command line is split into multiple lines and a continuation character, +, is added to all but the last line of the command.

### **CMDFilename**

Specifies the full path name of a file that will contain the commands specified with CMD. This parameter is optional. The maximum length of the file name is 1279 characters.

If you do not specify a file name, the MOVE MEDIA command will generate a default file name by appending the string *exec.cmds.media* to the Tivoli Storage Manager server directory. The server directory is the current working directory of the Tivoli Storage Manager server process.

The MOVE MEDIA command automatically allocates the file name specified or generated. If the file name exists, you can use the APPEND=YES parameter to add to the file. Otherwise, the file is overwritten. If a file is accidentally overwritten and you need to run the commands that had been in the file, issue the QUERY MEDIA command to rebuild the executable commands for the desired volumes. If the MOVE MEDIA command fails after the command file is allocated, the file is not deleted.

### **APPend**

Specifies to write at the beginning or ending of the command file data. The default is NO. Possible values are:

#### **No**

Specifies to write the data from the beginning of the command file. If the given command file exists, its contents are overwritten.

#### **Yes**

Specifies to append the command file by writing at the end of the command file data.

**CHECKLabel**

Specifies whether Tivoli Storage Manager should read volume labels for sequential media. For SCSI devices you can suppress label checking by setting the CHECKLabel to NO. This parameter is not applicable to 349X libraries. This parameter is optional. The default is NO. Possible values are:

**Yes**

Specifies that Tivoli Storage Manager attempts to read the media label. Reading the media label verifies that the correct volume is being checked out.

**No**

Specifies that Tivoli Storage Manager does not attempt to read media label. This increases performance because the read process does not occur.

**CAP**

Specifies which cartridge access port (CAP) to use for ejecting volumes if you specify REMOVE=YES. This parameter applies to volumes in ACSLS libraries only. If a CAP priority greater than zero does not exist in the library, this parameter is required. If a CAP priority greater than zero does exist in the library, this parameter is optional. If you do not specify the parameter, the ACSLS server will choose any available CAP with a priority greater than zero.

To display valid CAP identifiers (*x,y,z*), issue the QUERY CAP command with ALL specified from the Automated Cartridge System System Administrator (ACSSA) console on the ACSLS server host. The identifiers are as follows:

- x*      The Automated Cartridge System (ACS) ID. This identifier can be a number between 0-126.
- y*      The Library Storage Module (LSM) ID. This identifier can be a number between 0-23.
- z*      The CAP ID. This identifier can be a number between 0-11.

For more information, refer to your StorageTek documentation.

**Example: Move all full volumes out of the library**

Move all full volumes that are in the ARCHIVE sequential primary storage pool out of the library.

```
move media * stgpool=archive
```

**Example: Generate the checkin commands**

Generate the CHECKIN LIBVOLUME commands for full and partially full volumes that are in the ONSITE.ARCHIVE primary storage pool and stored in the overflow location, Room 2948/Bldg31.

MOVE MEDIA creates the executable commands in /tsm/move/media/checkin.vols

```
move media * stgpool=onsite.archive
wherestate=mountablenotinlib wherestatus=full,filling
ovflocation=room2948/bldg31
cmd="checkin libvol lib3494 &vol status=private"
cmdfilename=/tsm/move/media/checkin.vols
```

```
checkin libvolume lib3494 TAPE04 status=private
checkin libvolume lib3494 TAPE13 status=private
checkin libvolume lib3494 TAPE14 status=private
```

## MOVE MEDIA

**Tip:** Run the CHECKIN LIBVOLUME commands by issuing the MACRO command with the following as the macro name:

- /tsm/move/media/checkin.vols

### Related commands

*Table 194. Commands related to MOVE MEDIA*

| Command        | Description  |
|----------------|--|
| CANCEL PROCESS | Cancels a background server process.   |
| QUERY MEDIA    | Displays information about storage pool volumes moved by the MOVE MEDIA command. |
| QUERY PROCESS  | Displays information about background processes.                                 |

## MOVE NODEDATA (Move data by node in a sequential access storage pool)

Use this command to move data located in a sequential-access storage pool. You can move data for one or more nodes or for a group of colocated nodes. You can also move selected file spaces for a single node. The data can be located in a primary storage pool, a copy storage pool, or an active-data pool.

This command is helpful for reducing the number of volume mounts during client restore or retrieve operations by consolidating data for a specific node within a storage pool, or to move data to another storage pool. For example, you can use this command for moving data to a random-access storage pool in preparation for client restore processing.

Ensure that the access mode of the volumes *from which you are moving the node data* is read/write or read-only and that the access mode of the volumes *to which you are moving the node data* is set to read/write. This operation will not move data on volumes with access modes of offsite, unavailable, or destroyed.

The MOVE NODEDATA command takes two forms, depending on whether you are moving data only for selected filespace. The syntax and parameters for each form are defined separately.

**Restriction:** You cannot move node data into or out of a storage pool defined with a CENTERA device class.

Table 195. Commands related to MOVE NODEDATA

| Command             | Description   |
|---------------------|---|
| CANCEL PROCESS      | Cancels a background server process.  |
| COPY ACTIVATEDATA   | Copies active backup data.  |
| DEFINE COLLOCGROUP  | Defines a collocation group.  |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                      |
| DELETE COLLOCGROUP  | Deletes a collocation group.  |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                 |
| MOVE DATA           | Moves data from a specified storage pool volume to another storage pool volume. |
| QUERY ACTLOG        | Displays messages from the server activity log.                                 |
| QUERY COLLOCGROUP   | Displays information about collocation groups.                                  |
| QUERY FILESPACE     | Displays information about data in file spaces that belong to a client.         |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.     |
| QUERY OCCUPANCY     | Displays file space information by storage pool.                                |
| QUERY PROCESS       | Displays information about background processes.                                |
| QUERY STGPOOL       | Displays information about storage pools.                                       |
| QUERY VOLUME        | Displays information about storage pool volumes.                                |

## MOVE NODEDATA

*Table 195. Commands related to MOVE NODEDATA (continued)*

| Command            | Description                                     |
|--------------------|---|
| UPDATE COLLOCGROUP | Updates the description of a collocation group. |

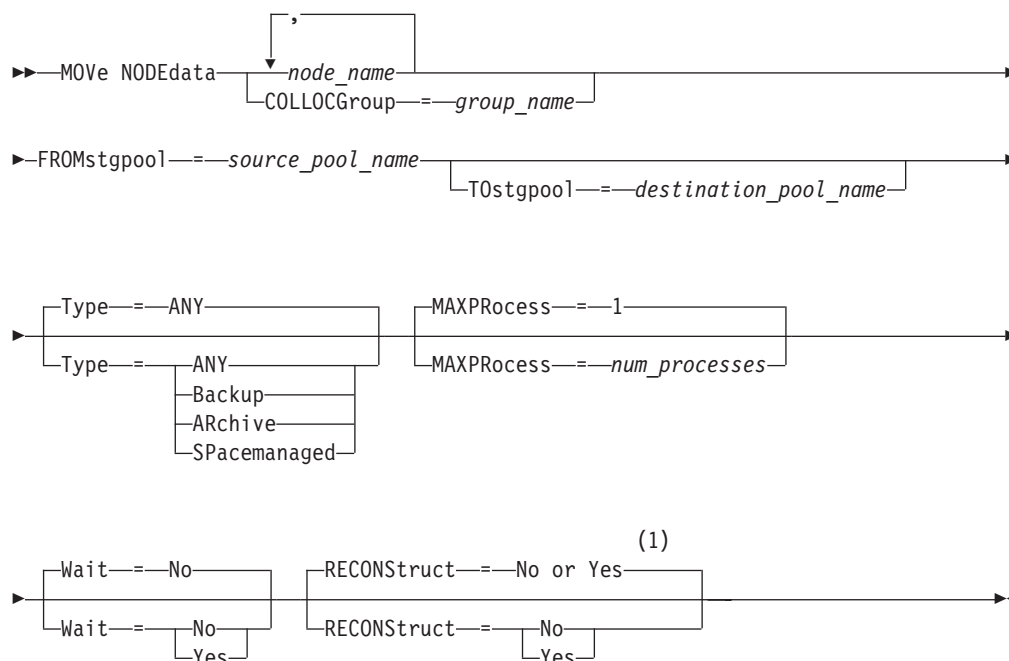
## MOVE NODEDATA (Move data in file spaces for one or more nodes or collocation group)

Use this command to move data in file spaces belonging to one or more nodes or to move data in file spaces belonging to a collocation group. A *collocation group* is a group of nodes whose data is collocated on a minimal number of sequential access volumes.

### Privilege class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the source storage pool. If your authorization is restricted storage privilege and you intend to move data to another storage pool, you must also have the appropriate authority for the destination storage pool.

### Syntax



### Notes:

- 1 The default is NO if either the source or target storage pool is random access. The default is YES if both the source and target storage pools are sequential access.

### Parameters

#### *node\_name* (Required unless the COLLOCGROUP parameter is specified)

Specifies the node name related to the data that is moved with this command. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names.

#### COLLOCGroup (Required unless the node\_name parameter is specified)

Specifies the name of the collocation group whose data is to be moved. Data for all the nodes belonging to the collocation group will be moved.

### **FROMstgpool (Required)**

Specifies the name of a sequential-access storage pool that contains data to be moved. This storage pool must be in the NATIVE or NONBLOCK data format.

### **TOstgpool**

Specifies the name of a storage pool to which data will be moved. This storage pool must be in the NATIVE or NONBLOCK data format. This parameter is optional and does not apply when the source storage pool is a copy storage pool or an active-data pool. That is, if the source storage pool is a copy storage pool the destination must be the same copy storage pool. Similarly, if the source storage pool is an active-data pool, the destination must be the same active-data pool. If a value is not specified, data is moved to other volumes within the source pool.

**Important:** If you are moving data within the same storage pool, there must be volumes available that do not contain the node data you are moving. That is, the server cannot use volumes that contain the data to be moved as destination volumes.

### **Type**

Specifies the type of files to be moved. This parameter is optional. The default value is ANY. If the source storage pool is an active-data pool, the only valid values are ANY and BACKUP. However, only the active versions of backup data are moved if TYPE=ANY. Possible values are:

#### **ANY**

Specifies that all types of files are moved.

#### **Backup**

Specifies that backup files are moved.

#### **ARchive**

Specifies that archive files are moved. This value is not valid for active-data pools.

#### **SPacemanaged**

Specifies that space-managed files (files that were migrated by a Tivoli Storage Manager for Space Management client) are moved. This value is not valid for active-data pools.

### **MAXPProcess**

Specifies the maximum number of parallel processes to use for moving data. This parameter is optional. You can specify a value from 1–999, inclusive. The default value is 1. Increasing the number of parallel processes should improve throughput.

When determining this value, consider the number of logical and physical drives that can be dedicated to this operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the move. Each process needs a mount point for storage pool volumes, and, if the device type is not FILE, each process also needs a drive.

### **Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default value is No. Possible values are:



**No**

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

The server displays messages that are created from the background process either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If a background process is canceled, some files may have already been moved before the cancellation.

**Yes**

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

**RECONSTRUCT**

Specifies whether to reconstruct file aggregates during data movement. Reconstruction removes empty space that has accumulated during deletion of logical files from an aggregate. This parameter is optional. If both the source and target storage pools are sequential access, the default value is YES. If either the source or target storage pool is random access, the default is NO. Possible values are:

**No**

Specifies that reconstruction of file aggregates will not be performed during the move.

**Yes**

Specifies that reconstruction of file aggregates will be performed during the move. You may only specify this option when both the source and the target storage pools are sequential-access.

**Tip:** Reconstruction removes inactive backup files in active-data pools. Specifying RECONSTRUCT=NO when moving data from volumes in an active-data pool will prevent the inactive backup files from being removed.

**Example: Move a specific node's data from a tape storage pool to a disk storage pool**

Move all data belonging to node MARY that is stored in storage pool TAPEPOOL. Data should be moved to disk storage pool BACKUPPOOL.

```
move nodedata mary
  fromstgpool=tapepool tostgpool=backuppool
```

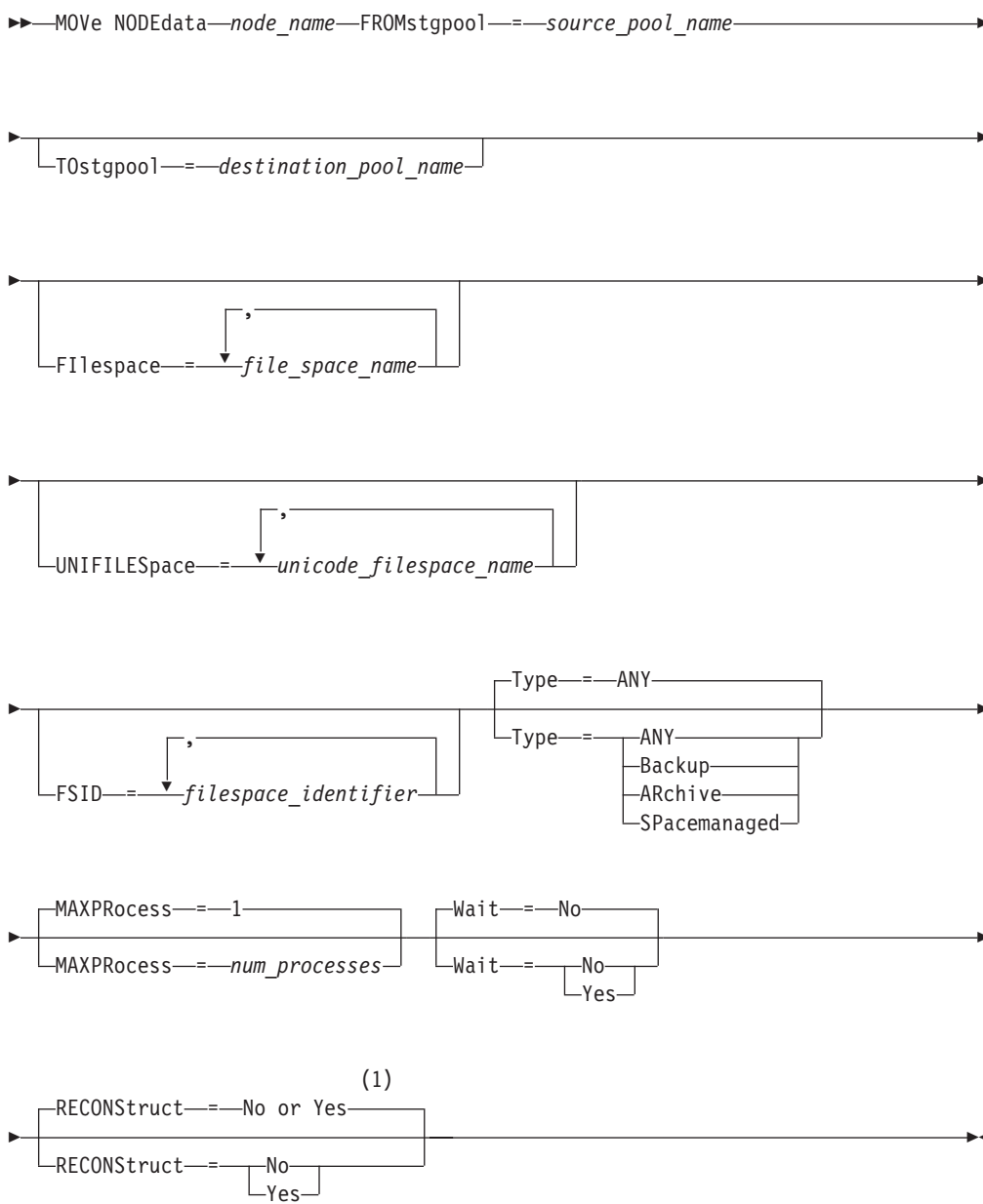
## MOVE NODEDATA (Move data from selected file spaces of a single node)

Use this command to move data for selected file spaces belonging to a single node.

### Privilege class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the source storage pool. If your authorization is restricted storage privilege and you intend to move data to another storage pool, you must also have the appropriate authority for the destination storage pool.

### Syntax



**Notes:**

- 1 The default is NO if either the source or target storage pool is random access. The default is YES if both the source and target storage pools are sequential access.

**Parameters*****node\_name* (Required)**

Specifies the node name related to the data that is moved with this command. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names.

**FROMstgpool (Required)**

Specifies the name of a sequential-access storage pool that contains data to be moved. This storage pool must be in the NATIVE or NONBLOCK data format.

**TOstgpool**

Specifies the name of a storage pool to which data will be moved. This storage pool must be in the NATIVE or NONBLOCK data format. This parameter is optional and does not apply when the source storage pool is a copy storage pool or an active-data pool. That is, if the source storage pool is a copy storage pool the destination must be the same copy storage pool. Similarly, if the source storage pool is an active-data pool, the destination must be the same active-data pool. If a value is not specified, data is moved to other volumes within the source pool.

**Important:** If you are moving data within the same storage pool, there must be volumes available that do not contain the node data you are moving. That is, the server cannot use volumes that contain the data to be moved as destination volumes.

**FILESspace**

Specifies the name of the non-Unicode filespace that contains data to be moved. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. This parameter is optional. If you do not specify a value for this parameter and values for UNIFILESPACE or the FSID or both, non-Unicode file spaces are not moved.

**UNIFILESpace**

Specifies the name of the Unicode filespace that contains data to be moved. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. This parameter is optional. If you do not specify a value for this parameter and values for FILESPACE or the FSID or both, non-Unicode file spaces are not moved.

**FSID**

Specifies file space identifiers (FSIDs) for the file spaces to be moved. Separate multiple names with commas and no intervening spaces. This parameter is optional.

**Type**

Specifies the type of files to be moved. This parameter is optional. The default value is ANY. If the source storage pool is an active-data pool, the only valid values are ANY and BACKUP. However, only the active versions of backup data are moved if TYPE=ANY. Possible values are:

**ANY**

Specifies that all types of files are moved.

**Backup**

Specifies that backup files are moved.

**ARchive**

Specifies that archive files are moved. This value is not valid for active-data pools.

**SPacemanaged**

Specifies that space-managed files (files that were migrated by a Tivoli Storage Manager for Space Management client) are moved. This value is not valid for active-data pools.

**MAXPRocess**

Specifies the maximum number of parallel processes to use for moving data. This parameter is optional. You can specify a value from 1–999, inclusive. The default value is 1. Increasing the number of parallel processes should improve throughput.

When determining this value, consider the number of logical and physical drives that can be dedicated to this operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the move. Each process needs a mount point for storage pool volumes, and, if the device type is not FILE, each process also needs a drive.

**Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default value is No. Possible values are:

**No**

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

The server displays messages that are created from the background process either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If a background process is canceled, some files may have already moved before the cancellation.

**Yes**

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

**RECONStruct**

Specifies whether to reconstruct file aggregates during data movement. Reconstruction removes empty space that has accumulated during deletion of logical files from an aggregate. This parameter is optional. If both the source and target storage pools are sequential access, the default value is YES. If either the source or target storage pool is random access, the default is NO. Possible values are:

**No**

Specifies that reconstruction of file aggregates will not be performed during the move.

**Yes**

Specifies that reconstruction of file aggregates will be performed during the move. You may only specify this option when both the source and the target storage pools are sequential-access.

**Tip:** Reconstruction removes inactive backup files in active-data pools. Specifying RECONSTRUCT=NO when moving data from volumes in an active-data pool will prevent the inactive backup files from being removed.

**Example: Move a node's non-Unicode and Unicode data**

Move data for node TOM in storage pool TAPEPOOL. Restrict movement of data to files in non-Unicode file spaces as well as Unicode file spaces, \\jane\d\$. Data should be moved to disk storage pool BACKUPPOOL.

```
move nodedata tom
  fromstgpool=tapepool tostgpool=backuppool
  filespace=* unifilespace=\\jane\d$
```

**Example: Move all node data from tape storage pools to a disk storage pool**

Move all data for node SARAH, from all primary sequential-access storage pools (for this example, TAPEPOOL\*) to DISKPOOL. To obtain a list of storage pools that contain data for node SARAH, issue either of the following QUERY OCCUPANCY or SELECT commands:

```
query occupancy sarah
SELECT * from OCCUPANCY where node_name='sarah'
```

**Attention:** For this example assume that the results were TAPEPOOL1, TAPEPOOL4, and TAPEPOOL5.

```
move nodedata sarah
  fromstgpool=tapepool1 tostgpool=DISKPOOL

move nodedata sarah
  fromstgpool=tapepool4 tostgpool=DISKPOOL

move nodedata sarah
  fromstgpool=tapepool5 tostgpool=DISKPOOL
```

**Example: Move a node's non-Unicode and Unicode file spaces**

The following is an example of moving non-Unicode and Unicode file spaces for a node. For node NOAH move non-Unicode file space \\servtuc\d\$ and Unicode file space \\tsmserv1\e\$ that has a file space ID of 2 from sequential access storage pool TAPEPOOL to random access storage pool DISKPOOL.

```
move nodedata noah
  fromstgpool=tapepool tostgpool=diskpool
  filespace=\\tsmserv1\d$ fsid=2
```

## NOTIFY SUBSCRIBERS (Notify managed servers to update profiles)

Use this command on a configuration manager to notify one or more managed servers to request that their configuration information be immediately refreshed.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶—NOTIfy SUBSCRIBers—PROFIle—=—profile_name—▶▶

```

### Parameters

#### PROFIle (Required)

Specifies the name of the profile. Any managed servers that subscribe to the profile are notified. You can use wildcard characters to specify multiple profiles. To specify multiple profiles, separate the names with commas and no intervening spaces.

### Example: Notify managed servers to update profiles

Notify all managed servers that subscribe to a profile named DELTA to request updated configuration information.

```
notify subscribers profile=delta
```

### Related commands

Table 196. Commands related to NOTIFY SUBSCRIBERS

| Command             | Description  |
|---------------------|--|
| DEFINE SUBSCRIPTION | Subscribes a managed server to a profile.  |
| DELETE SUBSCRIBER   | Deletes obsolete managed server subscriptions.                                   |
| DELETE SUBSCRIPTION | Deletes a specified profile subscription.  |
| QUERY SUBSCRIBER    | Displays information about subscribers and their subscriptions to profiles.      |
| QUERY SUBSCRIPTION  | Displays information about profile subscriptions.                                |
| SET CONFIGMANAGER   | Specifies whether a server is a configuration manager.                           |
| SET CONFIGREFRESH   | Specifies a time interval for managed servers to contact configuration managers. |

## PARALLEL (Run multiple commands in a script in parallel)

Use this command to allow any following commands in the script to run in parallel. When a script starts, all commands will run serially until a PARALLEL command is encountered.

**Remember:** Parallel commands accessing common resources, such as tape drives, can run serially.

The script return code following a PARALLEL command will be the same as the script return code prior to the PARALLEL command. When a SERIAL command is encountered, the script return code will be set to the maximum return code from any previous commands run in parallel.

When invoking server commands that support the WAIT parameter after a PARALLEL command, the behavior is as follows:

- If you specify (or use the default) WAIT=NO, your script will not wait for the completion of the command when a subsequent SERIAL command is encountered and the return code from that command will reflect processing only up to the point that the command starts a background process. The final return code from the command will not be available to your script.
- If you specify WAIT=YES, your script will wait for the completion of the command when a subsequent SERIAL command is encountered and the return code from that command will reflect processing for the entire command.

In most cases, you should use WAIT=YES on commands run in parallel.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—PARALLEL—►►

### Parameters

None.

### Example: Write a script using the PARALLEL command

Write a script named *BACKUP* to back up two storage pools simultaneously. When both backups are complete, perform a snapshot database backup. Issue the run backup command to run the script.

```
/* set up for running parallel commands */
PARALLEL
/* backup two storage pools simultaneously */
BACKUP STGPPOOL PRIMPOOL1 COPYPOOL1 WAIT=YES
BACKUP STGPPOOL PRIMPOOL2 COPYPOOL2 WAIT=YES
/* wait for all previous commands to finish and set up
running serial commands*/
SERIAL
/* start snapshot db backup */
BACKUP DB DEVCLASS=TAPE TYPE=DBSNAPSHOT WAIT=YES
```

## PARALLEL

### Related commands

*Table 197. Commands related to PARALLEL*

| Command       | Description  |
|---------------|--|
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| RUN           | Runs a script.   |
| SERIAL        | Run commands in a script serially.                         |



## PING SERVER (Test the connection between servers)

Use this command to test the connection between the local server and a remote server.

**Important:** The name and password of the administrator client issuing this command must also be defined on the remote server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—PING SERVER—*server\_name*—◄◄

### Parameters

*server\_name* **(Required)**

Specifies the name of the remote server.

### Example: Ping a server

Test the connection to server FRED.

```
ping server fred
```

### Related commands

*Table 198. Commands related to PING SERVER*

| Command       | Description   |
|---------------|---|
| DEFINE SERVER | Defines a server for server-to-server communications. |
| QUERY SERVER  | Displays information about servers.                   |

## PREPARE (Create a recovery plan file)

Use this command to create a recovery plan file, which contains the information needed to recover a Tivoli Storage Manager server. You can store a recovery plan file on a file system that is accessible to the source Tivoli Storage Manager server or on a target Tivoli Storage Manager server.

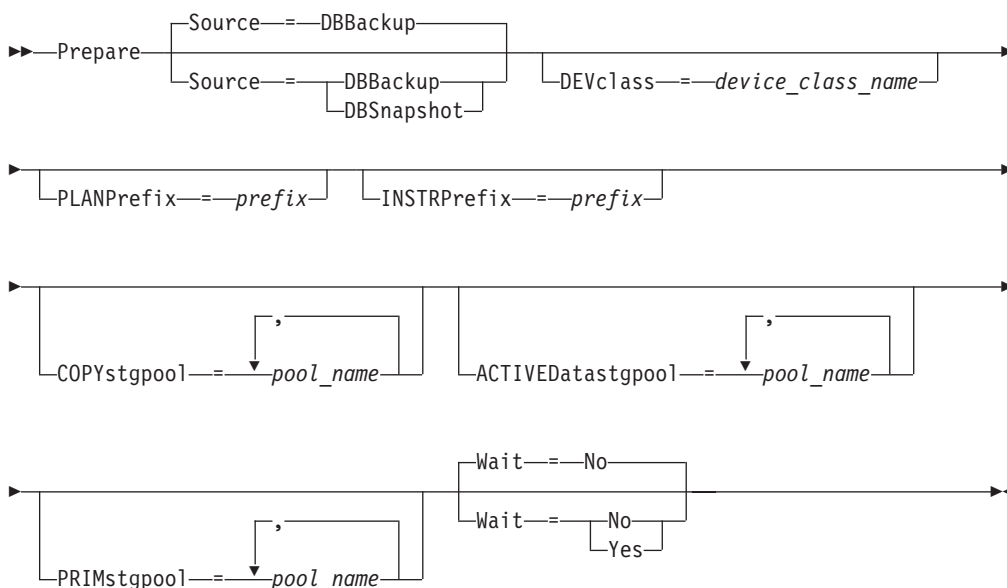
You can use the QUERY ACTLOG command to view whether the PREPARE command was successful.

You can also view this information from the server console or, if the WAIT parameter equals YES, an administrative client session.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### Source

Specifies the type of database backup series that Tivoli Storage Manager assumes when generating the recovery plan file. This parameter is optional. The default is DBBACKUP. The choices are:

#### DBBackup

Specifies that Tivoli Storage Manager assumes the latest full with plus database backup series.

#### DBSnapshot

Specifies that Tivoli Storage Manager assumes the latest database snapshot backup series.

#### DEVclass

Specifies the device class name that is used to create a recovery plan file object on a target server. The device class must have a device type of SERVER.

**Important:** The maximum capacity for the device class must be larger than the size of the recovery plan file. If the size of the recovery plan file exceeds the maximum capacity, the command fails.

The naming convention for the archive object that contains the recovery plan file on the target server is:

- **Filespace name:**  
ADSM.SERVER
- **High-level qualifier:**  
devclassprefix/servername.yyyymmdd.hhmmss
- **Low-level qualifier:**  
RPF.OBJ.1

The recovery plan file virtual volume name as recorded in the volume history table on the source server is in the format *servername.yyyymmdd.hhmmss*.

If the DEVCLASS parameter is not specified, the recovery plan file is written to a file based on the plan prefix.

If SOURCE=DBBACKUP is specified or is defaulted to, the volume history entry for the recovery plan file object specifies a volume type of RPFFILE. If SOURCE=DBSNAPSHOT is specified, the volume history entry specifies a volume type of RPFSNAPSHOT.

#### PLANPrefix

Specifies the path name prefix that is used in the recovery plan file name. The maximum length is 250 characters. This parameter is optional.

Tivoli Storage Manager appends to the prefix the sortable date and time format *yyymmdd.hhmmss*. For example: 20081115.051421.

The prefix can be one of the following:

- **Directory path:** End the prefix with the forward slash (/). For example:  
PLANPREFIX=/admsrv/recplans/

The resulting file name would look like this:  
/admsrv/recplans/20081115.051421

- **Directory path followed by a string:** Tivoli Storage Manager treats the string as part of the file name. For example:  
PLANPREFIX=/admsrv/recplans/accounting

The resulting file name looks like this:  
/admsrv/recplans/accounting.20081115.051421

Note the period before the date and time.

- **String only:** Tivoli Storage Manager specifies the directory path. Tivoli Storage Manager uses the name of the current working directory. For example, the current working directory is */opt/tivoli/tsm/server/bin* and you specify the following parameter:  
PLANPREFIX=shipping

The resulting file name looks like this:  
/opt/tivoli/tsm/server/bin/shipping.20081115.051421

Note the period before the date and time.

## PREPARE

If the PLANPREFIX parameter is not specified, Tivoli Storage Manager selects the prefix in one of these ways:

- If the SET DRMPLANPREFIX command has been issued, Tivoli Storage Manager uses the prefix specified in that command.
- If the SET DRMPLANPREFIX command has not been issued, Tivoli Storage Manager uses the directory path name of the current working directory. For example, the current working directory is the following:

```
/opt/tivoli/tsm/server/bin
```

The resulting file name looks like this:

```
/opt/tivoli/txm/server/bin/20081115.051421
```

### INSTRPrefix

Specifies the prefix of the path name used by Tivoli Storage Manager to locate the files that contain the recovery instructions. The maximum length is 250 characters.

The prefix can be one of the following:

- **Directory path:** End the prefix with the forward slash (/). For example:

```
INSTRPREFIX=/admsrv/recinstr/  
/admsrv/recinstr/RECOVERY.INSTRUCTIONS.GENERAL
```

- **Directory path followed by a string:** Tivoli Storage Manager treats the string as part of the file name. For example:

```
INSTRPREFIX=/admsrv/recinstr/accounts
```

Tivoli Storage Manager appends the appropriate recovery plan file stanza name. For the RECOVERY.INSTRUCTIONS.GENERAL file, the resulting file name is:

```
/admsrv/recinstr/accounts.RECOVERY.INSTRUCTIONS.GENERAL
```

- **String only:** Tivoli Storage Manager specifies the directory path and appends the appropriate recovery plan file stanza name. Tivoli Storage Manager uses the name of the current working directory. For example, the current working directory is */usr/lpp/admserv/bin* and you specify the following parameter:

```
INSTRPREFIX=shipping
```

For the RECOVERY.INSTRUCTIONS.GENERAL file, the resulting file name looks like this:

```
/opt/tivoli/tsm/server/bin/shipping.RECOVERY.INSTRUCTIONS.GENERAL
```

**If you do not specify the INSTRPREFIX parameter,** Tivoli Storage Manager selects the prefix in one of these ways:

- If the SET DRMINSTRPREFIX command has been issued, Tivoli Storage Manager uses the prefix specified in that command.
- If the SET DRMINSTRPREFIX command has not been issued, Tivoli Storage Manager uses the current working directory.

For example, if the current working directory is */opt/tivoli/tsm/server/bin*, for the RECOVERY.INSTRUCTIONS.GENERAL file, the resulting file name would be:

```
/opt/tivoli/tsm/server/bin/RECOVERY.INSTRUCTIONS.GENERAL
```

### PRIMstgpool

Specifies the names of the primary storage pools that you want to restore. Separate the storage pool names with commas and no intervening spaces. You

can use wildcard characters. If this parameter is not specified, Tivoli Storage Manager selects the storage pools as follows:

- If the SET DRMPRIMSTGPOOL command has been issued, Tivoli Storage Manager includes the primary storage pools named in that command.
- If the SET DRMPRIMSTGPOOL command has not been issued, Tivoli Storage Manager includes all the primary storage pools.

### **COPYstgpool**

Specifies the names of the copy storage pools used to back up the primary storage pools that you want to restore (see the PRIMSTGPOOL parameter). Separate storage pool names with commas and no intervening spaces. You can use wildcard characters. If this parameter is not specified, Tivoli Storage Manager selects the storage pools as follows:

- If the SET DRMCOPYSTGPOOL command has been issued, Tivoli Storage Manager includes those copy storage pools.
- If the SET DRMCOPYSTGPOOL command has not been issued, Tivoli Storage Manager includes all copy storage pools.

### **ACTIVEDatastgpool**

Specifies the names of the active-data storage pools that you want to have available for offsite access. Separate active-data storage-pool names with commas and no intervening spaces. You can use wildcard characters. If this parameter is not specified, Tivoli Storage Manager selects the storage pools as follows:

- If the SET ACTIVEDATASTGPOOL command has been previously issued with valid active-data storage pool names, Tivoli Storage Manager processes those storage pools.
- If the SET ACTIVEDATASTGPOOL command has not been issued, or all of the active-data storage pools have been removed using the SET ACTIVEDATASTGPOOL command, Tivoli Storage Manager processes only the active-data pool volumes that were marked on-site at the time the PREPARE command is run. Tivoli Storage Manager will mark these volumes as UNAVAILABLE.

### **Wait**

Specifies whether this command is processed in the background or foreground.

#### **No**

Specifies background processing. This is the default.

#### **Yes**

Specifies foreground processing.

You cannot specify YES from the server console.

## **Example: Create a recovery plan file**

Issue the PREPARE command and query the activity log to check the results.

```
prepare
query actlog search=prepare
```

## PREPARE

```
05/03/2008 12:01:13 ANR0984I Process 3 for PREPARE started in the
BACKGROUND at 12:01:13.
05/03/2008 12:01:13 ANR6918W PREPARE: Recovery instructions file
/home/guest/drmtest/prepare/tserver/DSM1509/
RECOVERY.INSTRUCTIONS.DATABASE not found.
05/03/2008 12:01:13 ANR6918W PREPARE: Recovery instructions file
/home/guest/drmtest/prepare/tserver/DSM1509/
RECOVERY.INSTRUCTIONS.STGPOOL not found.
05/03/2008 12:01:13 ANR6913W PREPARE: No volumes with backup data
exist in copy storage pool CSTORAGEP.
05/03/2008 12:01:13 ANR6913W PREPARE: No volumes with backup data
exist in copy storage pool CSTORAGEPSM.
05/03/2008 12:01:14 ANR6920W PREPARE: Generated replacement volume
name BACK4X@ is not valid for device type
8MM. Original volume name: BACK4X. Stanza is
PRIMARY.VOLUMES.REPLACEMENT macro.
05/03/2008 12:01:14 ANR6900I PREPARE: The recovery plan file
/home/guest/drmtest/prepare/plandir/DSM1509/
r.p.20080503.120113 was created.
05/03/2008 12:01:14 ANR0985I Process 3 for PREPARE running in the
BACKGROUND completed with completion state
SUCCESS at 12:01:14.
```

### Related commands

Table 199. Commands related to PREPARE

| Command                  | Description   |
|--------------------------|---|
| DELETE VOLHISTORY        | Removes sequential volume history information from the volume history file.           |
| QUERY DRMSTATUS          | Displays DRM system parameters.   |
| QUERY RPFCONTENT         | Displays the contents of a recovery plan file.  |
| QUERY RPFILE             | Displays information about recovery plan files.                                       |
| QUERY SERVER             | Displays information about servers.   |
| QUERY VOLHISTORY         | Displays sequential volume history information that has been collected by the server. |
| SET DRMACTIVEDATASTGPOOL | Specifies that active-data storage pools are managed by DRM.                          |
| SET DRMCOPYSTGPOOL       | Specifies that copy storage pools are managed by DRM.                                 |
| SET DRMINSTRPREFIX       | Specifies the prefix portion of the path name for the recovery plan instructions.     |
| SET DRMPPLANVPOSTFIX     | Specifies the replacement volume names in the recovery plan file.                     |
| SET DRMPPLANPREFIX       | Specifies the prefix portion of the path name for the recovery plan.                  |
| SET DRMPRIMSTGPOOL       | Specifies that primary storage pools are managed by DRM.                              |
| SET DRMRPFEXPIREDAYS     | Set criteria for recovery plan file expiration.                                       |
| UPDATE VOLHISTORY        | Adds or changes location information for a volume in the volume history file.         |

## QUERY commands

Use the QUERY command to request or display information about an IBM Tivoli Storage Manager object.

The following is a list of QUERY commands for Tivoli Storage Manager:

- “QUERY ACTLOG (Query the activity log)” on page 587
- “QUERY ADMIN (Display administrator information)” on page 592
- “QUERY ASSOCIATION (Query client node associations with a schedule)” on page 596
- “QUERY AUDITOCUPANCY (Query client node storage utilization)” on page 598
- “QUERY BACKUPSET (Query a backup set)” on page 601
- “QUERY BACKUPSETCONTENTS (Query contents of a backup set)” on page 606
- “QUERY CLOPTSET (Query a client option set)” on page 609
- “QUERY COLLOCGROUP (Query a collocation group)” on page 611
- “QUERY CONTENT (Query the contents of a storage pool volume)” on page 614
- “QUERY COPYGROUP (Query copy groups)” on page 621
- “QUERY DATAMOVER (Display data mover definitions)” on page 626
- “QUERY DB (Display database information)” on page 629
- “QUERY DBSPACE (Display database storage space)” on page 632
- “QUERY DEVCLASS (Display information on one or more device classes)” on page 633
- “QUERY DIRSPACE (Query storage utilization of FILE directories)” on page 637
- “QUERY DOMAIN (Query a policy domain)” on page 638
- “QUERY DRIVE (Query information about a drive)” on page 641
- “QUERY DRMEDIA (Query disaster recovery media)” on page 645
- “QUERY DRMSTATUS (Query disaster recovery manager system parameters)” on page 654
- “QUERY ENABLED (Query enabled events)” on page 657
- “QUERY EVENT (Query scheduled and completed events)” on page 659
- “QUERY EVENTRULES (Query rules for server or client events)” on page 671
- “QUERY EVENTSERVER (Query the event server)” on page 673
- “QUERY EXPORT (Query for active or suspended export operations)” on page 674
- “QUERY FILESPACE (Query one or more file spaces)” on page 681
- “QUERY LIBRARY (Query a library)” on page 686
- “QUERY LIBVOLUME (Query a library volume)” on page 689
- “QUERY LICENSE (Display license information)” on page 692
- “QUERY LOG (Display information on the recovery log)” on page 693
- “QUERY MACHINE (Query machine information)” on page 695
- “QUERY MEDIA (Query sequential access storage pool media)” on page 698
- “QUERY MGMTCLASS (Query a management class)” on page 704
- “QUERY MOUNT (Display information on mounted sequential access volumes)” on page 707

## QUERY commands

- “QUERY NASBACKUP (Query NAS backup images)” on page 708
- “QUERY NODE (Query nodes)” on page 712
- “QUERY NODEDATA (Query client data in volumes)” on page 719
- “QUERY NODEGROUP (Query a node group)” on page 722
- “QUERY OCCUPANCY (Query client file spaces in storage pools)” on page 724
- “QUERY OPTION (Query server options)” on page 728
- “QUERY PATH (Display a path definition)” on page 729
- “QUERY POLICYSET (Query a policy set)” on page 733
- “QUERY PROCESS (Query one or more server processes)” on page 736
- “QUERY PROFILE (Query a profile)” on page 738
- “QUERY PROXYNODE (Query proxy authority for a client node)” on page 741
- “QUERY RECOVERYMEDIA (Query recovery media)” on page 742
- “QUERY REQUEST (Query one or more pending mount requests)” on page 745
- “QUERY RESTORE (Query restartable restore sessions)” on page 746
- “QUERY RPFCONTENT (Query recovery plan file contents stored on a target server)” on page 749
- “QUERY RPFFILE (Query recovery plan file information stored on a target server)” on page 751
- “QUERY SAN (Query the devices on the SAN)” on page 754
- “QUERY SCHEDULE (Query schedules)” on page 757
- “QUERY SCRIPT (Query Tivoli Storage Manager scripts)” on page 765
- “QUERY SERVER (Query a server)” on page 768
- “QUERY SERVERGROUP (Query a server group)” on page 772
- “QUERY SESSION (Query client sessions)” on page 774
- “QUERY SHREDSTATUS (Query shredding status )” on page 778
- “QUERY SPACETRIGGER (Query the space triggers)” on page 780
- “QUERY STATUS (Query system parameters)” on page 782
- “QUERY STGPOOL (Query storage pools)” on page 788
- “QUERY SUBSCRIBER (Display subscriber information)” on page 799
- “QUERY SUBSCRIPTION (Display subscription information)” on page 801
- “QUERY SYSTEM (Query the system configuration and capacity)” on page 803
- “QUERY TAPEALERTMSG (Display status of SET TAPEALERTMSG command)” on page 805
- “QUERY TOC (Display table of contents for a backup image)” on page 806
- “QUERY VIRTUALFSMAPPING (Query a virtual file space mapping)” on page 809
- “QUERY VOLHISTORY (Display sequential volume history information)” on page 811
- “QUERY VOLUME (Query storage pool volumes)” on page 818



## QUERY ACTLOG (Query the activity log)

Use this command to display messages generated by the server. This command offers you various filtering options that can be used to limit the amount of messages displayed and the time it takes to process this query. If you do not specify any parameters with this command, all messages generated in the last hour are displayed.

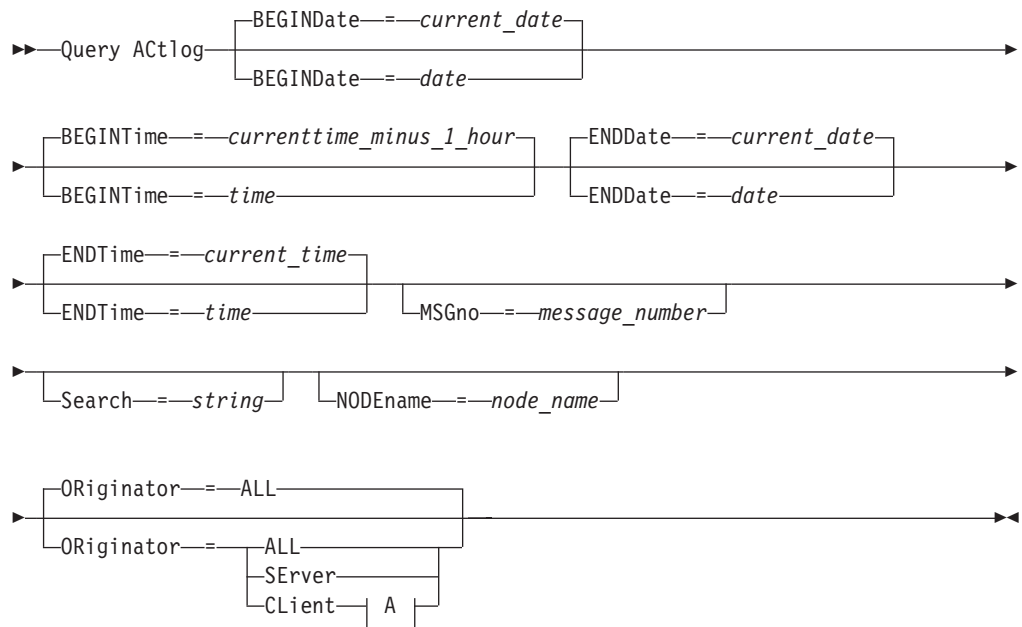
The activity log contains all messages that are sent to the server console under normal operation. The results of commands entered at the server console are not recorded in the activity log, unless the command affects or starts a background process or client session. Error messages are displayed in the activity log.

**Note:** You cannot schedule the QUERY ACTLOG command using the DEFINE SCHEDULE command.

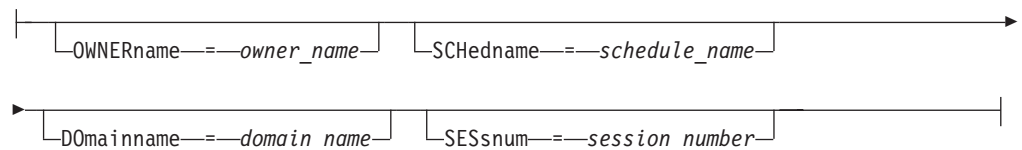
### Privilege class

Any administrator can issue this command.

### Syntax



**A:**



### Parameters

#### BEGINDate

Specifies the beginning date of the range for messages to be displayed. All

messages meeting the time range criteria that occurred after this date are displayed. The default is the current date. This parameter is optional.

You can specify the date using one of the following values:

| Value                  | Description  | Example  |
|------------------------|--|--|
| <i>MM/DD/YYYY</i>      | A specific date  | 09/15/1998   |
| TODAY                  | The current date   | TODAY  |
| TODAY-days or<br>-days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -7 or -7.<br><br>To display information beginning with messages created a week ago, you can specify <code>BEGINDATE=TODAY-7</code> or <code>BEGINDATE= -7</code> . |

### BEGINTime

Specifies the beginning time of the range for messages to be displayed. All messages meeting the time range criteria that occurred after this time are displayed. If you do not specify time, all messages that occurred in the last hour are displayed.

You can specify the time using one of the following values:

| Value                  | Description  | Example   |
|------------------------|--|---|
| <i>HH:MM:SS</i>        | A specific time on the specified begin date                          | 10:30:08  |
| NOW                    | The current time on the specified begin date                         | NOW   |
| NOW+HH:MM or<br>+HH:MM | The current time plus hours and minutes on the specified begin date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 with <code>BEGINTIME=NOW+3</code> or <code>BEGINTIME=+3</code> , Tivoli Storage Manager displays messages with a time of 12:00 or later on the begin date.                                |
| NOW-HH:MM or<br>-HH:MM | The current time minus hours and minutes on the specified begin date | NOW-04:00 or -04:00.<br><br>If you issue the <code>QUERY ACTLOG</code> command at 9:00 with <code>BEGINTime=NOW-3:30</code> or <code>BEGINTime= -3:30</code> , Tivoli Storage Manager displays messages with a time of 5:30 or later on the begin date. |

### ENDDate

Specifies the ending date of the range for messages to be displayed. All messages meeting the time range criteria that occurred before this date are displayed. If you do not specify a value, the current date is used. This parameter is optional.

You can specify the date using one of the following values:

| Value             | Description      | Example    |
|-------------------|------------------|------------|
| <i>MM/DD/YYYY</i> | A specific date  | 09/15/1998 |
| TODAY             | The current date | TODAY      |

| Value               | Description  | Example  |
|---------------------|--|--|
| TODAY-days or -days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY-1 or -1.<br><br>To display information created up to yesterday, you can specify ENDDATE=TODAY-1 or simply ENDDATE= -1. |

### ENDTime

Specifies the ending time of the range for messages to be displayed. All messages meeting this time range criteria that occurred before this time are displayed. If you do not specify a value, all messages are displayed up to the time when you issued this command. This parameter is optional.

You can specify the time using one of the following values:

| Value               | Description  | Example   |
|---------------------|--|---|
| HH:MM:SS            | A specific time on the specified end date                          | 10:30:08  |
| NOW                 | The current time on the specified end date                         | NOW   |
| NOW+HH:MM or +HH:MM | The current time plus hours and minutes on the specified end date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 with ENDTIME=NOW+3:00 or ENDTIME= +3:00, Tivoli Storage Manager displays messages with a time of 12:00 or earlier on the end date you specify.    |
| NOW-HH:MM or -HH:MM | The current time minus hours and minutes on the specified end date | NOW-03:30 or -03:30.<br><br>If you issue this command at 9:00 with ENDTIME=NOW-3:30 or ENDTIME= -3:30, IBM Tivoli Storage Manager displays messages with a time of 5:30 or earlier on the end date you specify. |

### MSGno

Specifies an integer that defines the number of the message to be displayed from the activity log. This is just the numeric part of the message. This parameter is optional.

### Search

Specifies a text string that you want to search for in the activity log. Enclose the string expression in quotation marks if it contains blanks. You can use text and a wildcard character to specify this string. This parameter is optional.

**Note:** Do not enter as a text string either the IBM Tivoli Storage Manager server name or text and a wildcard character that would find the server name. If you do so the output will include messages that do not include the search string.

### NODENAME

Specifies that the query displays messages logged for this node. If you do not specify a value for this parameter, messages for all nodes are displayed.

### ORiginator

Specifies that the query displays messages logged by the server, client, or both. The default is ALL. Possible values are:

#### ALL

Specifies that the query displays messages that originated from the client and the server.

#### SErver

Specifies that the query displays messages that originated from the server.

#### CLient

Specifies that the query displays messages that originated from the client.

You can specify one of the following values to minimize processing time when querying the activity log for messages logged by the client:

#### OWNERname

Specifies that the query displays messages logged for a particular owner. If you do not specify a value for this parameter, messages for all owners are displayed.

#### SCHedname

Specifies that the query displays messages logged by a particular scheduled client activity. If you do not specify a value for this parameter, messages for all schedules are displayed.

#### DOmainname

Specifies that the query displays messages logged for a particular policy domain to which a named schedule belongs. This parameter is optional, unless you are specifying a schedule name.

#### SESSnum

Specifies that the query displays messages logged from a particular client session number. If you do not specify a value for this parameter, messages for all client sessions are displayed.

### Example: Search activity log for messages with specific text

Search the activity log for any message that contains the string “delete”. The output includes only messages produced during the past hour. Issue the command:

```
query actlog search=delete
```

| Date/Time           | Message  |
|---------------------|--|
| 08/27/1998 15:19:43 | ANR0812I Inventory client file expiration complete: 0 files deleted. |

### Example: Search activity log for messages within a specific time frame

Display messages that occurred yesterday between 9:30 and 12:30. Issue the command:

```
query actlog begindate=today-1
begintime=09:30:00 endtime=12:30:00
```

| Date/Time           | Message   |
|---------------------|---|
| 10/21/1998 10:52:36 | ANR0407I Session 3921 started for administrator ADMIN (WebBrowser) (HTTP 9.115.20.100(2315)). |
| 10/21/1998 11:06:08 | ANR0405I Session 3922 ended for administrator ADMIN (WebBrowser).                             |
| 10/21/1998 12:16:50 | ANR0405I Session 3934 ended for administrator ADMIN (WebBrowser).                             |
| 10/21/1998 12:36:18 | ANR2501I Schedule AGADM_TEST deleted from policy domain STANDARD.                             |

### Example: Search activity log for messages from specific client node

Search the activity log for Tivoli Storage Manager messages from the client for node JEE. Issue the command:

```
query actlog originator=client node=jee
```

| Date/Time           | Message   |
|---------------------|---|
| 06/10/1998 15:46:22 | ANE4007E (Session No: 3 Node: JEE) Error processing '/jee/report.out': access to the object is denied |
| 06/11/1998 15:56:56 | ANE4009E (Session No: 4 Node: JEE) Error processing '/jee/work.lst': disk full condition              |

### Example: Search activity log for messages from specific client node and session

Search the activity log for Tivoli Storage Manager messages from the client for node FRED associated with Session 4. Issue the command:

```
query actlog search="(SESSION:4)"
```

| Date/Time           | Message   |
|---------------------|---|
| 09/23/2003 10:26:38 | ANR0406I Session 4 started for node FRED (WinNT) (Tcp/Ip colind(2463)). (SESSION: 4)                                |
| 09/23/2003 10:26:40 | ANR8493I FILE volume C:\CODE\522\00000000.BFS mounted in drive OUTFILE4 (FILE) in library OUTFILE. (SESSION: 4)     |
| 09/23/2003 10:26:40 | ANR8340I FILE volume C:\CODE\522\00000000.BFS mounted. (SESSION:4)  |
| 09/23/2003 10:26:40 | ANR8468I FILE volume C:\CODE\522\00000000.BFS dismounted from drive OUTFILE4 (FILE) in library OUTFILE. (SESSION:4) |
| 09/23/2003 10:26:40 | ANR0403I Session 4 ended for node FRED (WinNT). (SESSION:4)   |

### Related commands

Table 200. Command related to QUERY ACTLOG

| Command             | Description   |
|---------------------|---|
| SET ACTLOGRETENTION | Specifies the number of days to retain log records in the activity log. |

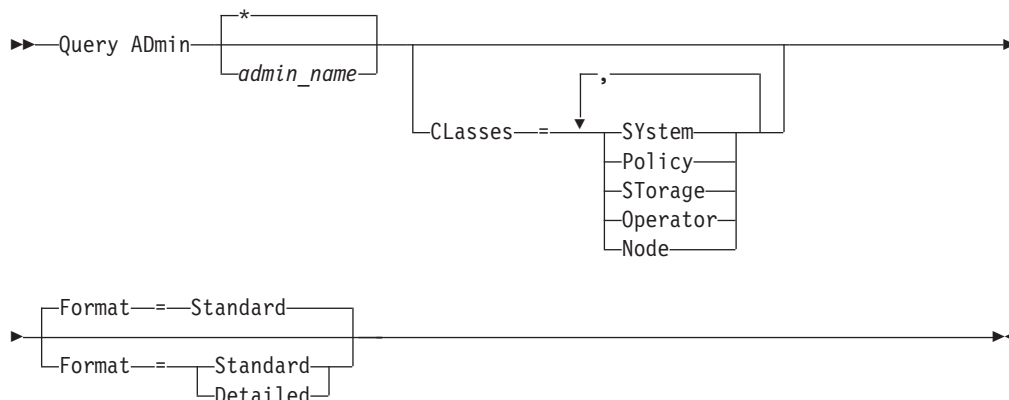
## QUERY ADMIN (Display administrator information)

Use this command to display information about one or more administrators.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *admin\_name*

Specifies the name of the administrator for which you want to display information. This parameter is optional. You can use wildcard characters to specify this name. If you do not specify a value for this parameter, all administrators are displayed.

#### **Classes**

Specifies that you want to restrict output to those administrators that have privilege classes that you specify. This parameter is optional. You can specify multiple privilege classes in a list by separating the names with commas and no intervening spaces. If you do not specify a value for this parameter, information on all administrators is displayed, regardless of privilege class. Possible values are:

##### **System**

Display information on administrators with system privilege.

##### **Policy**

Display information on administrators with policy privilege.

##### **STorage**

Display information on administrators with storage privilege.

##### **Operator**

Display information on administrators with operator privilege.

##### **Node**

Display information on users with client node privilege.

#### **Format**

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.

## Example: Display information about all administrators

Display partial information on all administrators. Issue the command:

```
query admin
```

| Administrator<br>Name | Days Since<br>Last Access | Days Since<br>Password<br>Set | Locked? | Privilege Classes |
|-----------------------|---------------------------|-------------------------------|---------|-------------------|
| -----                 | -----                     | -----                         | -----   | -----             |
| ADMIN                 | <1                        | <1                            | No      | System            |
| SERVER_CONSOLE        |                           |                               | No      | System            |

See “Field descriptions” for field descriptions.

## Example: Display complete information about one administrator

From a managed server, display complete information for the administrator named ADMIN. Issue the command:

```
query admin admin format=detailed
```

```
Administrator Name: ADMIN
Last Access Date/Time: 1998.06.04 17.10.52
Days Since Last Access: <1
Password Set Date/Time: 1998.06.04 17.10.52
Days Since Password Set: 26
Invalid Sign-on Count: 0
Locked?: No
Contact:
System Privilege: Yes
Policy Privilege: **Included with system privilege**
Storage Privilege: **Included with system privilege**
Operator Privilege: **Included with system privilege**
Client Access Privilege: **Included with system privilege**
Client Owner Privilege: **Included with system privilege**
Registration Date/Time: 05/09/1998 23:54:20
Registering Administrator: SERVER_CONSOLE
Managing profile:
Password Expiration Period: 90 Day (s)
Email Address:
```

See “Field descriptions” for field descriptions.

## Field descriptions

### 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.

**Invalid Sign-on Count**

Specifies the number of invalid sign-on attempts that have been made since the last successful sign-on. This count can only be non-zero when an invalid password limit (SET INVALIDPWLIMIT) is greater than zero. When the number of invalid attempts equals the limit set by the SET INVALIDPWLIMIT command, the administrator is locked out of the system.

**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.

**Operator Privilege**

Specifies whether the administrator has been granted operator privilege.

**Client Access Privilege**

Specifies that client access authority has been granted to a user with node privilege.

**Client Owner Privilege**

Specifies that client owner authority has been granted to a user with node 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. If this field contains \$\$CONFIG\_MANAGER\$\$, the administrator is associated with a profile that is managed by the configuration manager.

**Managing Profile**

Specifies the profiles to which the managed server subscribed to get the definition of this administrator.

**Password Expiration Period**

Specifies the administrator's password expiration period.



**Email Address**

Applies only to IBM Tivoli Storage Manager Express only and is shown for compatibility purposes.

**Related commands**

*Table 201. Commands related to QUERY ADMIN*

| Command            | Description  |
|--------------------|--|
| GRANT AUTHORITY    | Assigns privilege classes to an administrator.   |
| QUERY NODE         | Displays partial or complete information about one or more clients.                            |
| QUERY STATUS       | Displays the settings of server parameters, such as those selected by the SET commands.        |
| REGISTER ADMIN     | Defines a new administrator without granting administrative authority.                         |
| REMOVE ADMIN       | Removes an administrator from the list of registered administrators.                           |
| RENAME ADMIN       | Changes an IBM Tivoli Storage Manager administrator's name.                                    |
| RESET PASSEXP      | Resets the password expiration for nodes or administrators.                                    |
| REVOKE AUTHORITY   | Revokes one or more privilege classes or restricts access to policy domains and storage pools. |
| SET INVALIDPWLIMIT | Sets the number of invalid logon attempts before a node is locked.                             |
| SET MINPWLENGTH    | Sets the minimum length for client passwords.  |
| SET PASSEXP        | Specifies the number of days after which a password is expired and must be changed.            |

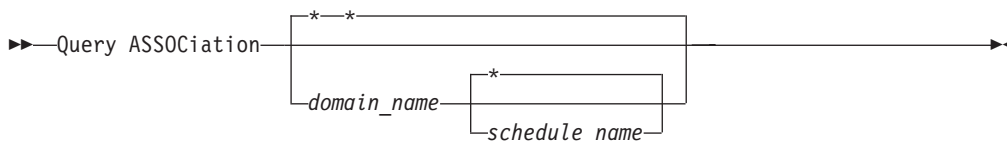
### QUERY ASSOCIATION (Query client node associations with a schedule)

Use this command to display information about which client nodes are associated with one or more schedules. Client nodes associated with a schedule perform operations such as backup or archive according to that schedule.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domain\_name*

Specifies the name of the policy domain to display. You can use a wildcard character 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.

##### *schedule\_name*

Specifies the name of the schedule to display. You can use a wildcard character 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.

#### Example: Display client nodes that are associated with a schedule

Display all the client nodes that are associated with each schedule that belongs to the EMPLOYEE\_RECORDS policy domain. Issue the command:

```
query association employee_records *
```

```

Policy Domain Name: EMPLOYEE_RECORDS
Schedule Name: WEEKLY_BACKUP
Associated Nodes: JOE JOHNSON LARRY SMITH SMITHERS TOM

```

See “Field descriptions” for field descriptions.

#### Field descriptions

##### Policy Domain Name

Specifies the name of the policy domain to which the schedule belongs.

##### Schedule Name

Specifies the name of the schedule.

##### Associated Nodes

Specifies the names of the client nodes that are associated with the specified schedule.

**Related commands***Table 202. Commands related to QUERY ASSOCIATION*

| Command            | Description   |
|--------------------|---|
| DEFINE ASSOCIATION | Associates clients with a schedule.                     |
| DELETE ASSOCIATION | Deletes the association between clients and a schedule. |

## QUERY AUDITOCUPANCY (Query client node storage utilization)

Use this command to display information about client node server storage utilization. To display current license audit information from the server, use the AUDIT LICENSE command prior to issuing the QUERY AUDITOCUPANCY command.

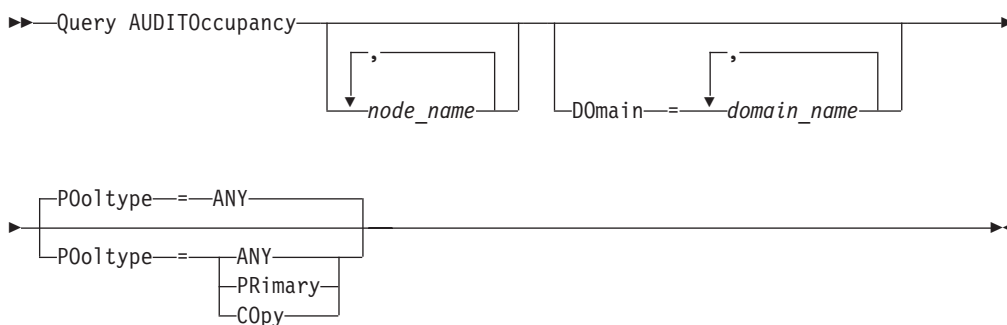
As part of a license audit operation, the server calculates, by node, the amount of backup, archive, and space management storage in use. For servers managing large amounts of data, this calculation can take a great deal of CPU time and can stall other server activity. You can use the AUDITSTORAGE server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

You can use the information from this query to determine if and where client node storage utilization needs to be balanced. This information can also assist you with billing clients for storage usage.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies a list of nodes for which to display server storage use information. Specify more than one node by separating the node names with commas, with no intervening spaces. You can use wildcard characters to specify names. The default (\*) is to query all client nodes. Use the **DOMAIN** parameter to limit this list by policy domain. This parameter is optional.

#### **DOMAIN**

Specifies a list of policy domains to restrict which nodes are displayed. Nodes belonging to the specified policy domains are displayed. Specify more than one policy domain by separating the policy domain names with commas, with no intervening spaces. You can use wildcard characters to specify names. This parameter is optional.

#### **POoltype**

Specifies the type of storage pool to display. This parameter is optional. The default is ANY. Possible values are:

**ANY**

Specifies both primary and copy storage pools. The value presented is the total for the two pools.

**PRimary**

Specifies primary storage pools only.

**COpy**

Specifies copy storage pools only.

**Example: Display storage usage**

Display combined storage use in primary and copy storage pools. Issue the command:

query auditoccupancy

License information as of last audit on 05/22/1996 14:49:51.

| Client Node Name | Backup<br>Storage<br>Used (MB) | Archive<br>Storage<br>Used (MB) | Space-Managed<br>Storage Used<br>(MB) | Total<br>Storage<br>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                           |

See “Field descriptions” for field descriptions.

**Field descriptions**

**Client Node Name**

Specifies the name of the client node.

**Backup Storage Used (MB)**

Specifies, in megabytes, the total backup storage use for the node.

**Archive Storage Used (MB)**

Specifies, in megabytes, the total archive storage use for the node.

**Space-Managed Storage Used (MB)**

Specifies, in megabytes, the amount of server storage used to store files migrated from the client node by a Tivoli Storage Manager for Space Management client.

**Total Storage Used (MB)**

Specifies, in megabytes, the total storage use for the node.

**Related commands**

Table 203. Commands related to QUERY AUDIT OCCUPANCY

| Command        | Description   |
|----------------|---|
| AUDIT LICENSES | Checks for compliance with defined licenses.  |
| QUERY LICENSE  | Displays information about licenses and audits.   |
| QUERY STATUS   | Displays the settings of server parameters, such as those selected by the SET commands. |

## QUERY AUDIT OCCUPANCY

*Table 203. Commands related to QUERY AUDIT OCCUPANCY (continued)*

| Command                | Description   |
|------------------------|---|
| REGISTER LICENSE       | Registers a new license with the IBM Tivoli Storage Manager server. |
| SET LICENSEAUDITPERIOD | Specifies the number of days between automatic license audits.      |

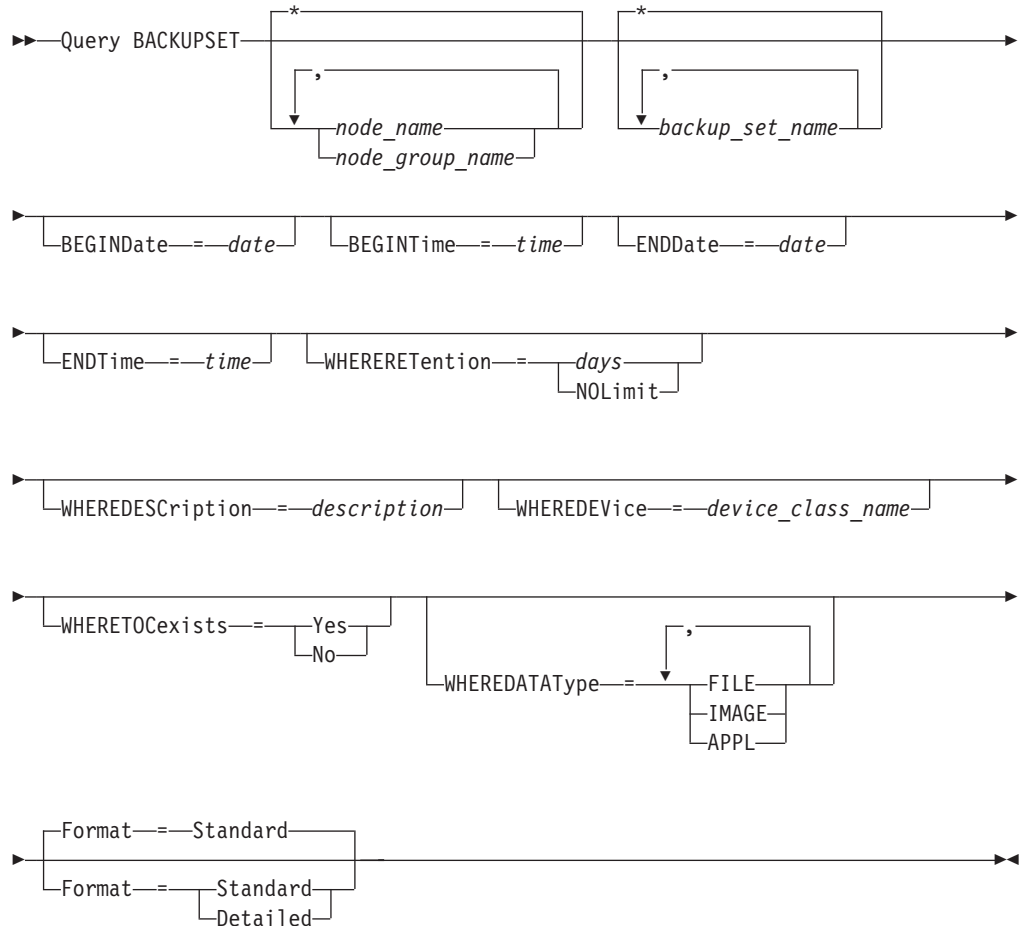
## QUERY BACKUPSET (Query a backup set)

Use this command to display information about one or more backup sets.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name* or *node\_group\_name* (Required)

Specifies the name of the client node and node groups whose data is contained in the backup set to be displayed. To specify multiple node names and node group names, separate the names with commas and no intervening spaces. You can use wildcard characters with node names but not with node group names.

#### *backup\_set\_name*

Specifies the name of the backup set whose information is to be displayed. The backup set name you specify can contain wildcard characters. You can specify more than one backup set name by separating the names with commas and no intervening spaces.

#### **BEGINDate**

Specifies the beginning date of the range in which the point-in-time date of the

## QUERY BACKUPSET

backup set to be displayed must fall. This parameter is optional. You can use this parameter with the **BEGINTIME** parameter to specify a range for the date and time. If you specify a begin date without a begin time, the time will be at 12:00 a.m. (midnight) on the date you specify.

You can specify the date by using one of the following values:

| Value                      | Description                            | Example         |
|----------------------------|--|-----------------|
| <i>MM/DD/YYYY</i>          | A specific date                        | 09/15/1999      |
| <i>TODAY</i>               | The current date                       | TODAY           |
| <i>TODAY+days or +days</i> | The current date plus days specified.  | TODAY +3 or +3. |
| <i>TODAY-days or -days</i> | The current date minus days specified. | TODAY -3 or -3. |

### BEGINTime

Specifies the beginning time of the range in which the point-in-time date of the backup set to be displayed must fall. This parameter is optional. You can use this parameter with the **BEGINDATE** parameter to specify a range for the date and time. If you specify a begin time without a begin date, the date will be the current date at the time you specify.

You can specify the time by using one of the following values:

| Value                      | Description  | Example              |
|----------------------------|--|----------------------|
| <i>HH:MM:SS</i>            | A specific time                                    | 10:30:08             |
| <i>NOW</i>                 | The current time                                   | NOW                  |
| <i>NOW+HH:MM or +HH:MM</i> | The current time plus hours and minutes specified  | NOW+02:00 or +02:00. |
| <i>NOW-HH:MM or -HH:MM</i> | The current time minus hours and minutes specified | NOW-02:00 or -02:00. |

### ENDDate

Specifies the ending date of the range in which the point-in-time date of the backup set to be displayed must fall. This parameter is optional. You can use this parameter with the **ENDTIME** parameter to specify an ending date and time. If you specify an end date without an end time, the time will be at 11:59:59 p.m. on the specified end date.

You can specify the date by using one of the following values:

| Value                      | Description   | Example         |
|----------------------------|---|-----------------|
| <i>MM/DD/YYYY</i>          | A specific date   | 09/15/1999      |
| <i>TODAY</i>               | The current date  | TODAY           |
| <i>TODAY+days or +days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3. |
| <i>TODAY-days or -days</i> | The current date minus days specified.  | TODAY -3 or -3. |

### ENDTime

Specifies the ending time of the range in which the point-in-time date of the backup set to be displayed must fall. This parameter is optional. You can use



this parameter with the ENDDATE parameter to specify a date and time. If you specify an end time without an end date, the date will be the current date at the time you specify.

You can specify the time by using one of the following values:

| Value                                  | Description  | Example              |
|--|--|----------------------|
| <i>HH:MM:SS</i>                        | A specific time                                    | 10:30:08             |
| NOW                                    | The current time                                   | NOW                  |
| NOW+ <i>HH:MM</i><br>or + <i>HH:MM</i> | The current time plus hours and minutes specified  | NOW+02:00 or +02:00. |
| NOW- <i>HH:MM</i><br>or - <i>HH:MM</i> | The current time minus hours and minutes specified | NOW-02:00 or -02:00. |

#### WHERERetention

Specifies the retention value, specified in days, that must be associated with the backup sets to be displayed. You can specify an integer from 0 to 30000. The values are:

*days*

Specifies that backup sets that are retained this number of days are displayed.

#### NOLimit

Specifies that backup sets that are retained indefinitely are displayed.

#### WHEREDescription

Specifies the description that must be associated with the backup set to be displayed. The description you specify can contain wildcard characters. This parameter is optional. Enclose the description in quotation marks if it contains any blank characters.

#### WHEREDevice

Specifies the name of the device class that must be associated with the backup set to be displayed. You can use wildcard characters to specify a device class name. This parameter is optional.

#### WHEREToExists

Specifies whether a backup set must have a table of contents in order to be displayed. This parameter is optional. The default is to display all backup sets whether or not they have a table of contents.

#### WHEREDataType

Specifies the data type of backup set to be displayed. This parameter is optional. The default is to display all types of backup sets. To specify multiple data types, separate data types with commas and no intervening spaces.

#### FILE

Specifies that a file level backup set is to be displayed. File level backup sets contain files and directories backed up by the backup-archive client.

#### IMAGE

Specifies that an image backup set is to be displayed. Image backup sets contain images created by the backup-archive client BACKUP IMAGE command.

#### APPL

Specifies that an application backup set is to be displayed. Application backup sets contain data backed up to the Tivoli Storage Manager server by applications using the Tivoli Storage Manager application programming

## QUERY BACKUPSET

interface. This parameter applies only if the Tivoli Storage Manager server was upgraded from Tivoli Storage Manager Express and only for the Tivoli Storage Manager Express versions of Data Protection for Exchange and Data Protection for SQL.

### Format

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 backup sets.

#### Detailed

Specifies that complete information is displayed for the specified backup sets.

### Example: Query a backup set

Display information for backup sets whose names begin with PERS\_DATA. The backup sets belong to the node JANE and are assigned to the DVLMENT device class.

```
query backupset jane pers_data*
```

```
Node Name: JANE
Backup Set Name: PERS_DATA.3089
Data Type: File
Date/Time: 03/17/2007 16:17:47
Retention Period: 60
Device Class Name: DVLMENT
Description: backupset created from /srvr
Has Table of Contents (TOC)?: Yes
```

## Field descriptions

### Node Name

Specifies the name of the client node whose data is contained in the backup set.

### Backup Set Name

Specifies the name of the backup set.

### Data Type:

Displays the data type of the backup sets. Possible types are file, image, and application.

### Date/Time

Specifies the date and time (PITDate and PITTime) of the GENERATE BACKUPSET command. The PITDate and PITTime specify that files that were active on the specified date and time and that are still stored on the Tivoli Storage Manager server are to be included in the backup set, even if they are inactive at the time you issue the GENERATE BACKUPSET command. The default is the date on which the GENERATE BACKUPSET command is run.

### Retention Period

Specifies the number of days that the backup set is retained on the server.

### Device Class Name

Specifies the name of the device class for which the volumes containing the backup set is assigned.

**Description**

Specifies the description associated with the backup set.

**Has Table of Contents (TOC)?**

Specifies whether the backup set has a table of contents.

**Related commands**

*Table 204. Commands related to QUERY BACKUPSET*

| Command                 | Description   |
|-------------------------|---|
| DEFINE BACKUPSET        | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP        | Defines a group of nodes.                               |
| DEFINE NODEGROUPMEMBER  | Adds a client node to a node group.                     |
| GENERATE BACKUPSET      | Generates a backup set of a client's data.              |
| GENERATE BACKUPSETTOC   | Generates a table of contents for a backup set.         |
| DELETE BACKUPSET        | Deletes a backup set.                                   |
| DELETE NODEGROUP        | Deletes a node group.                                   |
| DELETE NODEGROUPMEMBER  | Deletes a client node from a node group.                |
| QUERY BACKUPSETCONTENTS | Displays contents contained in backup sets.             |
| QUERY NODEGROUP         | Displays information about node groups.                 |
| UPDATE BACKUPSET        | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP        | Updates the description of a node group.                |

## QUERY BACKUPSETCONTENTS (Query contents of a backup set)

Use this command to display information about the files and directories contained in a backup set for a client node.

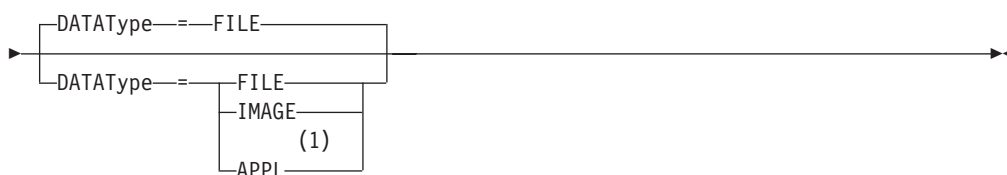
**Remember:** Processing this command can use considerable network resources and mount points.

## Privilege class

To issue this command, you must have system privilege or policy privilege for the domain to which the client node is assigned.

## Syntax

►►—Query BACKUPSETCONTENTS—*node name*—*backup set name*————►



**Notes:**

- 1 The APPL parameter is supported only if Tivoli Storage Manager was upgraded from Tivoli Storage Manager Express™.

## Parameters

*node\_name* (Required)

Specifies the name of the client node whose data is contained in the backup set to display. The name you specify cannot contain wildcard characters nor can it be a list of node names separated by commas.

*backup\_set\_name* **(Required)**

Specifies the name of the backup set to display. The name that you specify cannot contain wildcard characters nor can it be a list of node names that are separated by commas.

## DATAType

Specifies that the backup set containing the specified types of data is to be queried. This parameter is optional. The default is that a file level backup set is to be queried. Possible values are:

## FILE

Specifies that a file level backup set is to be queried. File level backup sets contain files and directories backed up by the backup-archive client.

## IMAGE

Specifies that an image backup set is to be queried. Image backup sets contain images created by the backup-archive client **BACKUP IMAGE** command.

## APPL

Specifies that an application backup set is to be queried. Application backup sets contain data backed up to the Tivoli Storage Manager Server.

by applications using the Tivoli Storage Manager application programming interface (including Data Protection clients).

**Important:** Application backup sets are supported only if Tivoli Storage Manager was upgraded from Tivoli Storage Manager Express.

## Example: Query contents of a backup set for a specific node

Display the contents from backup set named PERS\_DATA.3099 belonging to client node JANE. Issue the command:

```
query backupsetcontents jane pers_data.3099
```

| Node Name | Filespace Name | Client's Name for File |
|-----------|----------------|------------------------|
| JANE      | /srvr          | /deblock               |
| JANE      | /srvr          | /deblock.c             |
| JANE      | /srvr          | /dsmerror.log          |
| JANE      | /srvr          | /dsmxxxxx.log          |
| JANE      | ...            | .....                  |

## Field descriptions

### Node Name

Specifies the name of the client node whose data is contained in the backup set.

### Filespace Name

Specifies the name of the file space to which the specified file belongs.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (?), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

### Client's Name for File

Specifies the name of the file.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (?), blanks, unprintable characters, or "...". These

## QUERY BACKUPSETCONTENTS

characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

A file name that is displayed as "....." indicates that both the file path and file name were not successfully converted. An example of the path and name could be:

```
my\dir\...
```

### Related commands

*Table 205. Commands related to QUERY BACKUPSETCONTENTS*

| Command               | Description   |
|-----------------------|---|
| DEFINE BACKUPSET      | Defines a previously generated backup set to a server.  |
| GENERATE BACKUPSET    | Generates a backup set of a client's data.              |
| GENERATE BACKUPSETTOC | Generates a table of contents for a backup set.         |
| DELETE BACKUPSET      | Deletes a backup set.                                   |
| QUERY BACKUPSET       | Displays backup sets.                                   |
| UPDATE BACKUPSET      | Updates a retention value associated with a backup set. |

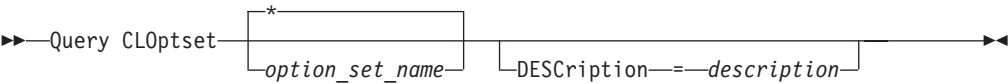
## QUERY CLOPTSET (Query a client option set)

Use this command to query a client option set.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

*option\_set\_name*

Specifies the name of the client option set to be queried. You can use wildcard characters to specify this name. This parameter is optional. The default is option set names.

#### DESCRiption

Specifies the description used on the DEFINE or UPDATE CLOPTSET commands to be used as a filter. If the description contains spaces, enclose it in quotation marks. This parameter is optional.

### Example: Query a client option set

From a managed server, query a client option set named ENG. Issue the following command:

```
query cloptset eng
```

```

      Optionset:  ENG
      Description:
Last Update by (administrator): $$CONFIG_MANAGER$$
      Managing profile:

      Option: SCROLLINES
      Sequence number: 0
Use Option Set Value (FORCE): No
      Option Value: 40

      Option: SCROLLPROMPT
      Sequence number: 0
Use Option Set Value (FORCE): No
      Option Value: yes

```

### Field descriptions

#### Optionset

Specifies the name of the option set.

#### Description

Specifies the description of the client option set.

#### Last Update by (administrator)

Specifies the name of the administrator that most recently updated the option set. If this field contains \$\$CONFIG\_MANAGER\$\$, the client option set is associated with a profile that is managed by the configuration manager.

## QUERY CLOPTSET

### Managing Profile

Specifies the profile to which the managed server subscribed to get the definition of the client option set.

### Option

Specifies the name of the option.

### Sequence Number

Specifies the sequence number of the option.

### Use Option Set Value (FORCE)

Specifies whether the server option setting overrides the client's option setting. NO indicates the server option setting does not override the client option. YES indicates the server's option setting overrides the client option setting. This option is set with the **FORCE** parameter on the **DEFINE CLIENTOPT** command.

### Option Value

Specifies the value of the option.

## Related commands

*Table 206. Commands related to QUERY CLOPTSET*

| Command                | Description  |
|------------------------|--|
| COPY CLOPTSET          | Copies a client option set.  |
| DEFINE CLIENTOPT       | Adds a client option to a client option set.                           |
| DEFINE CLOPTSET        | Defines a client option set.   |
| DELETE CLIENTOPT       | Deletes a client option from a client option set.                      |
| DELETE CLOPTSET        | Deletes a client option set.   |
| UPDATE CLIENTOPT       | Updates the sequence number of a client option in a client option set. |
| UPDATE CLOPTSET        | Updates the description of a client option set.                        |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                     |



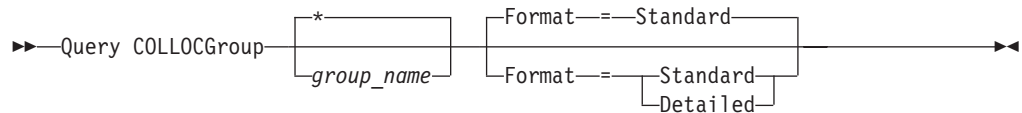
# QUERY COLLOCGROUP (Query a collocation group)

Use this command to display the collocation groups defined on the server.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

### group\_name

Specifies the name of the collocation group to display. To specify multiple names, use a wildcard character. This parameter is optional. The default is to display all collocation groups.

### Format

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.

#### Detailed

Specifies that complete information is displayed. To display the members of the collocation group, you must specify FORMAT=DETAILED.

## Example: Display defined collocation groups

Display the collocation groups defined on the server. Issue the following command:

```
query collocgroup
```

| Collocation Group Name | Collocation Group Description |
|------------------------|-------------------------------|
| DEPT_ED                | Education department          |
| GROUP1                 | Low cap client nodes.         |

See “Field descriptions” on page 612 for field descriptions.

## Example: Display detailed information for collocation groups

Display complete information about all collocation groups and determine which client nodes belong to which collocation groups. Issue the following command:

```
query collocgroup format=detailed
```

## QUERY COLLOCGROUP

```
Collocation Group Name: DEPT_ED
Collocation Group Description: Education department
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2004 10:59:03
Collocation Group Member(s): EDU_1 EDU_7

Collocation Group Name: GROUP1
Collocation Group Description: Low cap client nodes.
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2004 10:59:16
Collocation Group Member(s): CHESTER REX NOAH JARED
```

See “Field descriptions” for field descriptions.

### Field descriptions

#### Collocation Group Name

The name of the collocation group.

#### Collocation Group Description

The description for the collocation group.

#### Last Update by (administrator)

The name of the administrator that defined or most recently updated the collocation group.

#### Last Update Date/Time

The date and time that an administrator defined or most recently updated the collocation group.

#### Collocation Group Member(s)

The members of the collocation group.

### Related commands

Table 207. Commands related to QUERY COLLOCGROUP

| Command             | Description  |
|---------------------|--|
| DEFINE COLLOCGROUP  | Defines a collocation group.   |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                       |
| DEFINE STGPOOL      | Defines a storage pool as a named collection of server storage media.            |
| DELETE COLLOCGROUP  | Deletes a collocation group.   |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                  |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces.    |
| QUERY NODE          | Displays partial or complete information about one or more clients.              |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.      |
| QUERY STGPOOL       | Displays information about storage pools.  |
| REMOVE NODE         | Removes a client from the list of registered nodes for a specific policy domain. |
| UPDATE COLLOCGROUP  | Updates the description of a collocation group.                                  |
| UPDATE STGPOOL      | Changes the attributes of a storage pool.  |



## QUERY CONTENT (Query the contents of a storage pool volume)

Use this command to display information about files in a storage pool volume, as well as the names of client files that link to a deduplicated group of files.

You can use this command to identify files that the server has found to be damaged and files that have been backed up to a copy storage pool or copied to an active-data pool. This command is useful when a volume is damaged or before you:

- Request the server to fix inconsistencies between a volume and the database
- Move files from one volume to another volume
- Delete a volume from a storage pool

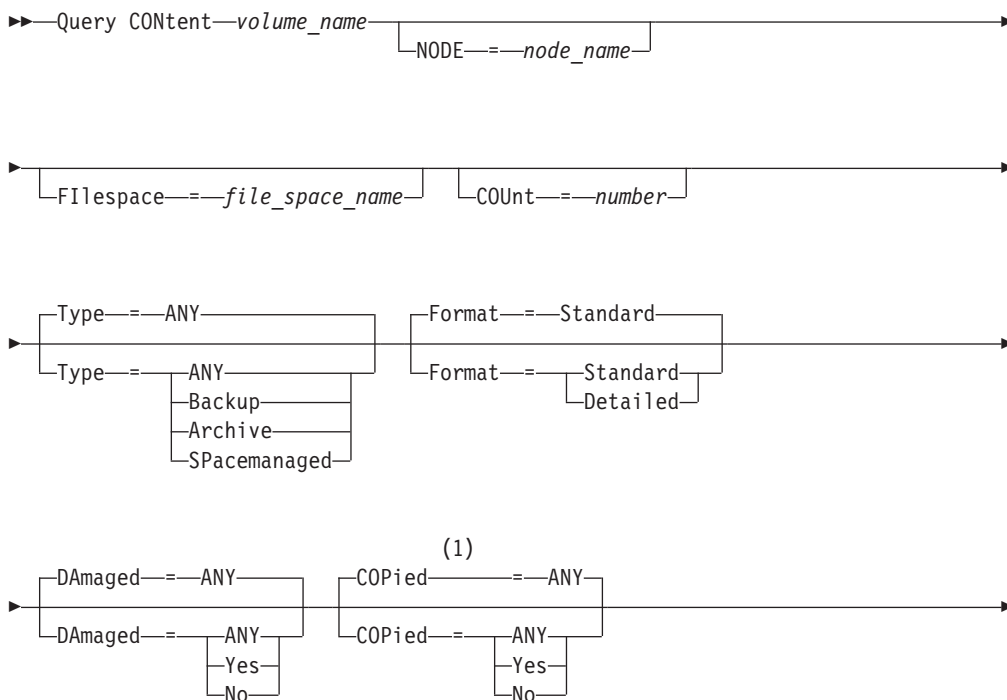
Because this command can take a long time to run and the results can be very large, consider using the **COUNT** parameter to limit the number of files displayed.

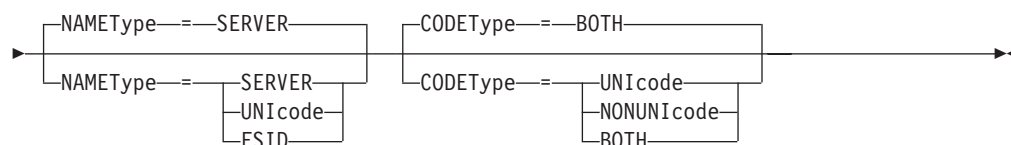
**Note:** Files that are cached in a disk volume and that are marked as damaged are not included in the results.

### Privilege class

Any administrator can issue this command.

### Syntax





### Notes:

- 1 Use this parameter only for volumes in primary storage pools.

## Parameters

### *volume\_name* (Required)

Specifies the volume to be queried.

### NODE

Specifies the backup-archive client or the Tivoli Storage Manager for Space Management associated with the file space to query. This parameter is optional. You can use wildcard characters to specify this name. If you do not specify a name, all backup-archive and Tivoli Storage Manager for Space Management clients are included.

### Filespace

Specifies the file space to query. This parameter is optional. You can use wildcard characters to specify this name. File space names are case-sensitive. If you do not specify a file space name, all file spaces are included.

For a server that has clients with Unicode support, you may need to have the server convert the file space name that you enter. For example, you may need to have the server convert the name you enter from the server's code page to Unicode. See the **NAMETYPE** parameter for details. If you do not specify a file space name or specify only a single wildcard character for the name, you can use the **CODETYPE** parameter to limit the operation to Unicode file spaces or non-Unicode file spaces.

### COUNt

Specifies the number of files to be displayed. This parameter is optional. You can specify either a positive integer or a negative 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

Specifies the types of files to query. This parameter is optional. The default value is ANY. If the volume being queried is assigned to an active-data pool, the only valid values are ANY and BACKUP. Possible values are:

#### ANY

Specifies that all types of files in the storage pool volume are queried: backup versions of files, archived copies of files, and files migrated by Tivoli Storage Manager for Space Management clients from client nodes.

#### Backup

Specifies that only backup files are queried.

#### Archive

Specifies that only archive files are queried. This value is not valid for active-data pools.

### **SPacemanaged**

Specifies that only space-managed files (files that were migrated by a Tivoli Storage Manager for Space Management client) are queried. This value is not valid for active-data pools.

### **Format**

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. Unicode names are converted to the server code page.

#### **Detailed**

Specifies that complete information is displayed. Unicode names are displayed in hexadecimal.

### **DAMaged**

Specifies criteria to restrict the query output based on whether files have been marked as damaged. For purposes of this criteria, the server examines only physical files (a file that may be a single logical file or an aggregate that consists of logical files). This parameter is optional. The default value is ANY. Possible values are:

#### **ANY**

Specifies that files are displayed regardless of whether the server has found the files to be damaged.

#### **Yes**

Specifies that only files that have been marked as damaged are displayed. These are files in which the server found errors when a user attempted to restore, retrieve, or recall the file, or when an AUDIT VOLUME command was run.

#### **No**

Specifies that only files *not* known to have been damaged are displayed.

### **COPIed**

Specifies criteria to restrict the query output based on whether files have been backed up to a copy storage pool. Whether files are stored in an active-data pool does not effect the output. 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 files displayed are only 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 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 files displayed are only 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.

### **NAMEType**

Specify how you want the server to interpret the file space names that you enter. This parameter is useful when the server has clients with Unicode support. A backup-archive client with Unicode support is currently available only for Windows, Macintosh OS 9, Macintosh OS X, and NetWare. Use this parameter only when you specify a partly or fully qualified file space name.

The default value is `SERVER`. Possible values are:

#### **SERVER**

The server uses the server's code page to interpret the file space names.

#### **UNICODE**

The server converts the file space names from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the names and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

#### **FSID**

The server interprets the file space names as their file space IDs (FSIDs).

### **CODEType**

Specify how you want the server to interpret the file space names that you enter. Use this parameter only when you enter a single wildcard character for the file space name.

The default value is `BOTH`, which means that the file spaces are included regardless of code page type. Possible values are:

#### **UNICODE**

Include only file spaces that are in Unicode.

#### **NONUNICODE**

Include only file spaces that are not in Unicode.

#### **BOTH**

Include file spaces regardless of code page type.

### **Example: Display the contents of a volume for a specific client node**

Query the contents of a volume and limit the results to files backed up from the PEGASUS client node.

For the volume `/tsmstg/diskvol1.dsm`, issue the command:

```
query content /tsmstg/diskvol1.dsm node=pegasus
type=backup
```

Results of the command include all logical files that make up any aggregate that is on the volume, even if the aggregate is stored on more than this volume. For aggregates, the query does not determine which logical files are actually stored on the volume for which the query is performed.

| Node Name | Type | Filespace     | FSID | Client's Name for File |
|-----------|------|---------------|------|------------------------|
| -----     | ---- | -----         | ---- | -----                  |
| PEGASUS   | Bkup | \\pegasus\e\$ | 1    | \\UNI_TEST\ SM01.DAT   |
| PEGASUS   | Bkup | \\pegasus\e\$ | 1    | \\UNI_TEST\ SM02.DAT   |

See “Field descriptions” on page 619 for field descriptions.

## **Example: Display detailed information for a tape volume**

Query the contents of the tape volume named WPD001. Display only files backed up by the node MARK and display only the first four files on the volume. Display complete information by issuing the following command:

```
query content wpd001 node=mark count=4 type=backup
format=detailed
```

```

Node Name: MARK
Type: Bkup
Filespace Name: \\mark\e$
Hexadecimal Filespace Name:
FSID: 1
Client's Name for File: \\UNI_TEST\ SM01.DAT
Hexadecimal Client's Name for File:
Aggregated?: 1/3
Stored Size: 2,746
Segment Number:
Cached Copy?: No

Node Name: MARK
Type: Bkup
Filespace Name: \\mark\e$
Hexadecimal Filespace Name:
FSID: 1
Client's Name for File: \\UNI_TEST\ SM02.DAT
Hexadecimal Client's Name for File:
Aggregated?: 2/3
Stored Size: 2,746
Segment Number:
Cached Copy?: No

Node Name: MARK
Type: Bkup
Filespace Name: \\mark\e$
Hexadecimal Filespace Name:
FSID: 1
Client's Name for File: \\UNI_TEST\ SM03.DAT
Hexadecimal Client's Name for File:
Aggregated?: 3/3
Stored Size: 2,746
Segment Number:
Cached Copy?: No

Node Name: MARK
Type: Bkup
Filespace Name: \\mark\fs$
Hexadecimal Filespace Name: 5c5c6d61726b5c6624
FSID: 2
Client's Name for File: \\UNI_TEST\ ELAP.REX
Hexadecimal Client's Name for File: 5c554e495f544553545c20454c41502e52
Aggregated?: 1/3
Stored Size: 2,372
Segment Number:
Cached Copy?: No
```

See “Field descriptions” on page 619 for field descriptions.



## Field descriptions

### Node Name

The node to which the file belongs.

**Type** The type of file: archive (Arch), backup (Bkup), or space-managed (SpMg) by a Tivoli Storage Manager for Space Management client.

### Filespace Name

The file space to which the file belongs.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

### Hexadecimal Filespace Name

The file space to which the file belongs. If the file space name is in Unicode, the name is displayed in hexadecimal format.

**FSID** The file space ID (FSID) for the file space. The server assigns a unique FSID when a file space is first stored on the server.

### Client's Name for File

The client's name for the file.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist.

### Hexadecimal Client's Name for File

The client's name for the file displayed in hexadecimal format.

### Aggregated?

Whether the file is a logical file that is stored as part of an aggregate. If the file is part of an aggregate, the sequence of this file within the aggregate and the total number of logical files in the aggregate are displayed. Results of the command include all logical files that make up any aggregate that is on the volume, even if the aggregate is stored on more than this volume. The query does not determine which logical files are actually stored on the volume for which the query is performed.

If the file is not part of an aggregate, the field simply displays "no."

## QUERY CONTENT

### Stored Size

The size of the physical file, in bytes. If the file is a logical file that is stored as part of an aggregate, this value indicates the size of the entire aggregate.

### Segment Number

For volumes in sequential-access storage pools, specifies whether the physical file (either a single logical file or an aggregate of logical files) is stored across multiple volumes. For example, if the logical file is stored in an aggregate that spans two volumes, the segment number indicates 1/2 (the first part of the physical file is stored on the volume) or 2/2 (the second part of the physical file is stored on the volume). If the segment number is 1/1, the physical file is completely stored on the volume. For volumes in random-access storage pools, no value is displayed for this field.

### Cached Copy?

Whether the physical file is a cached copy of a file migrated to the next storage pool. If the file is part of an aggregate, this value pertains to the aggregate.

## Related commands

*Table 208. Commands related to QUERY CONTENT*

| Command         | Description   |
|-----------------|---|
| BACKUP STGPOOL  | Backs up a primary storage pool to a copy storage pool.                                       |
| COPY ACTIVEDATA | Copies active backup data.  |
| DEFINE VOLUME   | Assigns a volume to be used for storage within a specified storage pool.                      |
| DELETE VOLUME   | Deletes a volume from a storage pool.   |
| RESTORE STGPOOL | Restores files to a primary storage pool from copy storage pools.                             |
| RESTORE VOLUME  | Restores files stored on specified volumes in a primary storage pool from copy storage pools. |
| UPDATE VOLUME   | Updates the attributes of storage pool volumes.   |

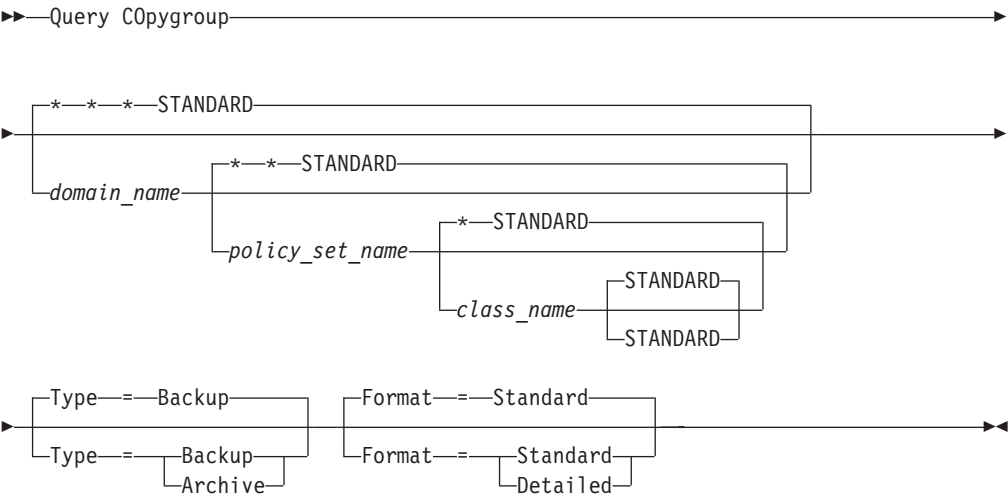
# QUERY COPYGROUP (Query copy groups)

Use this command to display information about one or more copy groups.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

*domain\_name*  
Specifies the policy domain associated with the copy group to query. This parameter is optional. You can use wildcard characters to specify names. 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.

*policy\_set\_name*  
Specifies the policy set associated with the copy group to query. This parameter is optional. You can use wildcard characters to specify names. 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.

*class\_name*  
Specifies the management class associated with the copy group to query. This parameter is optional. You can use wildcard characters to specify names. 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. The default is STANDARD.

## 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.

## QUERY COPYGROUP

### Archive

Specifies that you want to query archive copy groups.

### Format

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.

#### Detailed

Specifies that complete information is displayed.

### Example: Display detailed information on one backup copy group

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. Issue the command:

```
query copygroup employee_records vacation
activefiles format=detailed
```

```
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
Table of Contents (TOC) Destination:
Last Update by (administrator): ADMIN
Last Update Date/Time: 10/02/2002 17.51.49
```

### Example: Display information on the backup copy group in the STANDARD management class and policy set

From a managed server, display complete information on the backup copy group assigned to the STANDARD management class in the STANDARD policy set of the ADMIN\_RECORDS policy domain. Issue the command:

```
query copygroup admin_records
standard standard format=detailed
```

```

Policy Domain Name: ADMIN_RECORDS
Policy Set Name: STANDARD
Mgmt Class Name: STANDARD
Copy Group Name: STANDARD
Copy Group Type: Backup
Versions Data Exists: 2
Versions Data Deleted: 1
Retain Extra Versions: 30
Retain Only Version: 60
Copy Mode: Modified
Copy Serialization: Shared Static
Copy Frequency: 0
Copy Destination: BACKUPPOOL
Table of Contents (TOC) Destination:
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 2002.10.02 17.51.49
Managing profile: ADMIN_INFO
Changes Pending: Yes

```

### Example: Display information on an archive copy group

From a managed server, display complete information on the archive copy group STANDARD that is assigned to the MCLASS1 management class in the SUMMER policy set of the PROG1 policy domain. Issue the command:

```

query copygroup prog1 summer mclass1
type=archive format=detailed

```

```

Policy Domain Name: PROG1
Policy Set Name: SUMMER
Mgmt Class Name: MCLASS1
Copy Group Name: STANDARD
Copy Group Type: Archive
Retain Version: 730
Retention Initiation: Creation
Minimum Retention:
Copy Serialization: Shared Static
Copy Frequency: Cmd
Copy Mode: Absolute
Copy Destination: ARCHPOOL
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 2002.10.02 17.42.49
Managing profile: ADMIN_INFO

```

### Example: Display information on the copy group for a NAS backup

Query the copy group for the NAS backup. Issue the command:

```

query copygroup nasdomain
type=backup

```

```
Policy Domain Name: NASDOMAIN
Policy Set Name: ACTIVE
Mgmt Class Name: STANDARD
Copy Group Name: STANDARD
Copy Group Type: Backup
Versions Data Exists: 2
Versions Data Deleted: 1
Retain Extra Versions: 30
Retain Only Version: 60
Copy Mode: Modified
Copy Serialization: Shared Static
Copy Frequency: 0
Copy Destination: NASPOOL
Table of Contents (TOC) Destination: BACKUPPOOL
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 10/02/2002 12:16:52
Managing profile:
Changes Pending: Yes
```

### Field descriptions

#### Policy Domain Name

The name of the policy domain.

#### Policy Set Name

The name of the policy set.

#### Mgmt Class Name

The name of the management class.

#### Copy Group Name

The name of the copy group. This name is always STANDARD.

#### Copy Group Type

The type of the copy group.

#### Versions Data Exists

The maximum number of backup versions to retain for files that are currently on the client file system.

#### Versions Data Deleted

The maximum number of backup versions to retain for files that have been deleted from the client file system after being backed up using Tivoli Storage Manager.

#### Retain Extra Versions

The number of days to retain a backup version after that version becomes inactive.

#### Retain Only Versions

The number of days to retain the last backup version of a file that has been deleted from the client file system.

#### Retain Version

The number of days to keep an archive copy.

#### Retention Initiation

The time which the server initiates the retention time specified by the RETAIN VERSION parameter. CREATION specifies that the retention time is initiated at the time an archive copy is stored on the server. EVENT specifies that the retention time is initiated at the time the server is notified of a retention-initiating event for the archive copy.

**Minimum Retention**

The minimum number of days to keep an archive copy when Retention Initiation is EVENT. The value of this parameter is not displayed when Retention Initiation is CREATION.

**Copy Serialization**

Whether a file can be in use during an archive operation.

**Copy Frequency**

The copy frequency of the copy group. For archive copy groups, this value is always 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 is always ABSOLUTE.

**Copy Destination**

The name of the storage pool where the server initially stores files associated with this archive copy group.

**Table of Contents (TOC) Destination**

The name of the primary storage pool in which TOCs are initially stored for image backup operations in which TOC generation is requested.

**Last Update by (administrator)**

The name of the administrator or server that most recently updated the copy group. If this field contains \$\$CONFIG\_MANAGER\$\$, the copy group is associated with a domain that is managed by the configuration manager.

**Last Update Date/Time**

The date and time when the copy group was most recently defined or updated.

**Managing Profile**

The profile or profiles to which the managed server subscribed to get the definition of this policy copy group.

**Changes Pending**

Whether or not changes are being made but not activated. Once the changes are activated, the field resets to No.

**Related commands**

*Table 209. Commands related to QUERY COPYGROUP*

| Command          | Description  |
|------------------|--|
| DEFINE COPYGROUP | Defines a copy group for backup or archive processing within a specified management class. |
| DELETE COPYGROUP | Deletes a backup or archive copy group from a policy domain and policy set.                |
| UPDATE COPYGROUP | Changes one or more attributes of a copy group.  |

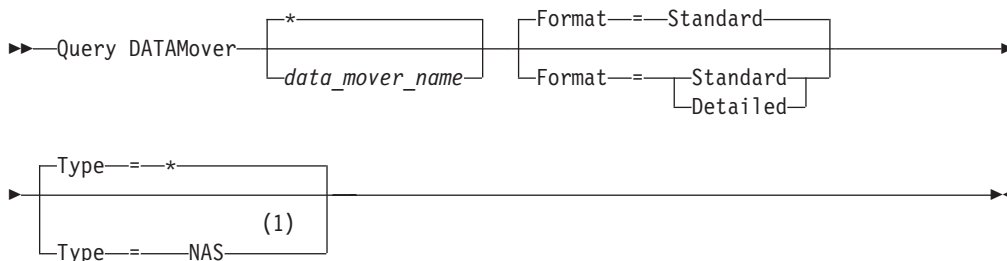
## QUERY DATAMOVER (Display data mover definitions)

Use this command to display data mover definitions.

### Privilege class

Any administrator can issue this command.

### Syntax



### Notes:

- 1 The type must be specified if FORMAT=DETAILED

### Parameters

#### *data\_mover\_name*

Specifies the name of the data mover to display. You can specify multiple names with a wildcard character. The default displays all data movers.

#### Format

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

##### Standard

Specifies that the assigned name and addressing information is specified.

##### Detailed

Specifies that complete information is displayed.

#### Type

Specifies the type of data mover to be displayed. This parameter must be specified if the FORMAT = DETAILED.

##### NAS

Specifies a NAS device.

### Example: Display information on all data movers

Display the data movers on the server. Issue the command:

```
query datamover
```

| Data Mover Name | Type | Online |
|-----------------|------|--------|
| NASMOVER1       | NAS  | Yes    |
| NASMOVER2       | NAS  | No     |

See “Field descriptions” on page 627 for field descriptions.



### Example: Display information for one data mover

Display partial information on data mover DATAMOVER6. Issue the command:

```
query datamover datamover6 type=nas
```

| Source Name | Type  | Online |
|-------------|-------|--------|
| -----       | ----- | -----  |
| DATAMOVER6  | NAS   | Yes    |

See “Field descriptions” for field descriptions.

### Example: Display detailed information on one data mover

Display detailed information on data mover DATAMOVER6. The TYPE parameter is required when FORMAT=DETAILED. Issue the command:

```
query datamover datamover6 format=detailed type=nas
```

```

Data Mover Name:  DataMover6
                  Type:      NAS
                  HLAddress:  9.115.38.5
                  LLAddress:  10000
                  USERid:    NDMPadmin
Storage Pool Data Format: NDMPDUMP
                  Online:    Yes
Last Update by (administrator): ADMIN
                  Last Update Date/Time: 04/28/2003 09:26:33

```

See “Field descriptions” for field descriptions.

### Field descriptions

#### Data Mover Name

Specifies the name of the data mover

**Type** Specifies the type of the data mover

#### HLAddress

Specifies the IP address for the data mover.

#### LLAddress

Specifies the TCP port number for the data mover.

#### User ID

Specifies the user ID that the server uses to get access to the data mover.

#### Storage Pool Data Format

Specifies the data format that is used by this data mover.

#### Online

Specifies the whether the data mover is online and available for use.

#### Last Update by (administrator)

Specifies the ID of the administrator who performed the last update.

#### Last Update Date/Time

Specifies the date and time when the last update occurred.

## QUERY DATAMOVER

### Related commands

*Table 210. Commands related to QUERY DATAMOVER*

| Command          | Description  |
|------------------|--|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server. |
| DELETE DATAMOVER | Deletes a data mover.  |
| UPDATE DATAMOVER | Changes the definition for a data mover.                       |

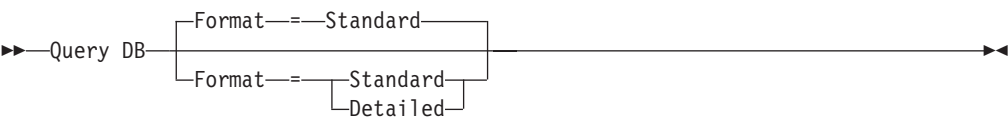
# QUERY DB (Display database information)

Use this command to display information about the database.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

### Format

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed.

#### Detailed

Specifies that complete information is displayed.

## Example: Display summary statistics about the database

Display statistical information about the database. Issue the command:

query db

| Database Name | Page Size(Bytes) | Total Pages | Usable Pages | Used Pages | Free Pages |
|---------------|------------------|-------------|--------------|------------|------------|
| TSMDB1        |                  | 32,776      | 32,504       | 24,220     | 8,284      |

See “Field descriptions” on page 630 for field descriptions.

## Example: Display detailed database information

Display detailed statistical information about the database. Issue the command:

query db format=detailed

```

Database Name : TSM_DB2
Total Size of File System(MB) : 1,748,800
Space Used by Database(MB) : 448
Free Space Available (MB) : 235,609
Page Size(Bytes) :
Total Pages : 32,776
Usable Pages : 32,504
Used Pages : 24,220
Free Pages : 8,284
Buffer Pool Hit Ratio : 99.3
Total Buffer Requests : 204,121
Sort Overflows : 0
Lock Escalation : 0
Package Cache Hit Ratio : 89.8
Last Database Reorganization : 05/25/2009 16:44:06
Full Device Class Name : FILE
Incrementals Since Last Full : 0
Last Complete Backup Date/Time: 05/18/2009 22:55:19

```

See “Field descriptions” for field descriptions.

## Field descriptions

### Database Name

The name of the database that is defined and configured for use by the Tivoli Storage Manager server.

### Total Size of File System(MB)

The total size, in megabytes, of the file systems in which the database is located.

### Space Used by Database(MB)

The amount of database space, in megabytes, that is in use.

### Free Space Available(MB)

The amount of database space, in megabytes, that is not in use.

### Page Size (Bytes)

The database page size in bytes.

### Total Pages

The total number of pages in the table space.

### Usable Pages

The number of usable pages in the table space.

### Used Pages

The number of used pages in the table space.

### Free Pages

The number of free pages in the table space.

### Buffer Pool Hit Ratio

The total hit ratio percent.

### Total Buffer Requests

The total number of buffer pool data logical reads and index logical reads since the last time the database was started or since the database monitor was reset.

### Sort Overflows

The total number of sorts that ran out of the sort heap and might have required disk space for temporary storage.

### Lock Escalation

The number of times that locks have been escalated from several row locks to a table lock.

### Package Cache Hit Ratio

A percentage indicating how well the package cache is helping to avoid reloading packages and sections for static SQL from the system catalogs. It also indicates how well the package cache is helping to avoid recompiling dynamic SQL statements. A high ratio indicates it is successful in avoiding these activities.

### Last Database Reorganization

The last time that the database manager performed an automatic reorganization activity.

### Full Device Class Name

The name of the device class this is used for full database backups.

### Incrementals Since Last Full

The number of incremental backups that were performed since the last full backup.

### Last Complete Backup Date/Time

The date and time of the last full backup.

## Related commands

Table 211. Commands related to QUERY DB

| Command        | Description  |
|----------------|--|
| BACKUP DB      | Backs up the IBM Tivoli Storage Manager database to sequential access volumes. |
| EXTEND DBSPACE | Adds directories to increase space for use by the database.                    |
| QUERY DBSPACE  | Displays information about the storage space defined for the database.         |

# QUERY DBSPACE (Display database storage space)

Use this command to display information about the directories used by the database to store data.

## Privilege class

Any administrator can issue this command.

## Syntax

►►—QUERY DBSpace—◄◄

## Parameters

None.

## Example: Display database storage space information

Display information about database storage space. Issue the command:

query dbspace

| Location  | Total Size of<br>File System (MB) | Space Used on<br>File System (MB) | Free Space<br>Available (MB) |
|-----------|-----------------------------------|-----------------------------------|------------------------------|
| /tsmdb001 | 1,748,800                         | 1,513,191.125                     | 117,804.422                  |
| /tsmdb002 | 1,748,800                         | 1,513,191.125                     | 117,804.422                  |

See “Field descriptions” for field descriptions.

## Field descriptions

### Location

Specifies the locations of database directories.

### Total Size of File System(MB)

The total size, in megabytes, of the file system in which the database is located.

### Space Used on Filesystem(MB)

The amount of database space, in megabytes, that is in use.

### Free Space Available(MB)

The amount of space, in megabytes, that is not in use.

## Related commands

Table 212. Commands related to QUERY DBSPACE

| Command        | Description  |
|----------------|--|
| BACKUP DB      | Backs up the IBM Tivoli Storage Manager database to sequential access volumes. |
| EXTEND DBSPACE | Adds directories to increase space for use by the database.                    |
| QUERY DB       | Displays allocation information about the database.                            |

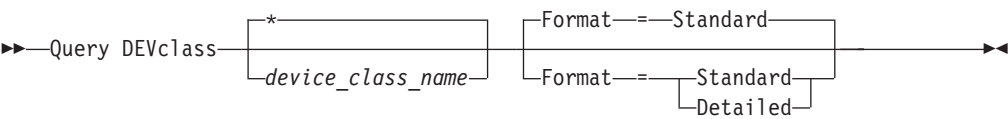
QUERY DEVCLASS (Display information on one or more device classes)

Use this command to display information on one or more device classes.

Privilege class

Any administrator can issue this command.

Syntax



Parameters

*device\_class\_name*  
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**  
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.

Example: List all device classes

Display information on all device classes.

query devclass

| 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           |
| 8MMSP2            | Sequential             | 2                  | 8MM         | DRIVE  | 44.4                  | DRIVES      |

See “Field descriptions” on page 634 for field descriptions.

Example: Display detailed information for a specific FILE device class

Display information in full detail on the PLAINFILES device class.

query devclass plainfiles format=detailed

## QUERY DEVCLASS

```
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:

Library:
Directory:
Server Name:
Retry Period:

Retry Interval:
Shared:
Last Update by (administrator): ADMIN
Last Update Date/Time: 05/31/2000 13:15:36
```

See “Field descriptions” for field descriptions.

### Example: Display detailed information for a specific 3592 device class

Display information in full detail on the 3592 device class.

```
query devclass 3592 format=detailed
```

```
Device Class Name: 3592
Device Access Strategy: Sequential
Storage Pool Count: 1
Device Type: 3592
Format: 3592
Est/Max Capacity (MB):
Mount Limit: DRIVES
Mount Wait (min): 60
Mount Retention (min): 60
Label Prefix: ADSM

Library: MANLIB
Directory:
Server Name:
Retry Period:

Retry Interval:
Shared:
HLAddr:
WORM: No
Scaled Capacity: 90
Drive Encryption: On
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 08/04/03 14:28:31
```

See “Field descriptions” for field descriptions.

### Field descriptions

#### Device Class Name

The name of the device class.

#### Device Access Strategy

How data is written to the device class.

#### Storage Pool Count

The number of storage pools that are assigned to the device class.



**Device Type**

The device type of the device class.

**Format**

The recording format.

**Est/Max Capacity (MB)**

The estimated or maximum capacity of a volume associated with the device class.

**Mount Limit**

The maximum number of sequential access volumes that can be mounted concurrently or specifies that DRIVES is the mount limit.

**Mount Wait (min)**

The maximum number of minutes to wait for a sequential access volume to be mounted.

**Mount Retention (min)**

The number of minutes to retain an idle sequential access volume before dismounting it.

**Label Prefix**

The high-level qualifier of the data set name that the server writes into the sequential access media labels.

**Library**

The name of the defined library object that contains the drives used by the device class.

**Directory**

The directory or directories for a shared FILE device class.

**Server Name**

The name of a defined server.

**Retry Period**

The interval over which the server should attempt to contact a target server if communications failure is suspected.

**Retry Interval**

How often the retries are done within a given retry period.

**Two-Sided**

Whether a removable file is two-sided.

**Shared**

Whether this FILE device class will be shared between the server and one or more storage agents.

**HLAddress**

The IP address of the device in dotted decimal format.

**WORM**

Whether this drive is a write once, read many (WORM) device.

**Scaled Capacity**

The percentage of the media capacity that can be used to store data.

**Drive Encryption**

Whether drive encryption is permitted. This field applies only to volumes in a storage pool associated with a device type of 3592 or LTO.

**Last Update by (administrator)**

The administrator that made the last update to the device class.

## QUERY DEVCLASS

### Last Update Date/Time

The date and time of the last update.

### Related commands

*Table 213. Commands related to QUERY DEVCLASS*

| Command         | Description   |
|-----------------|---|
| DEFINE DEVCLASS | Defines a device class.                               |
| DEFINE SERVER   | Defines a server for server-to-server communications. |
| DELETE DEVCLASS | Deletes a device class name.                          |
| QUERY DIRSPACE  | Displays information about FILE directories.          |
| QUERY SERVER    | Displays information about servers.                   |
| UPDATE DEVCLASS | Changes the attributes of a device class.             |

# QUERY DIRSPACE (Query storage utilization of FILE directories)

Use this command to display information about free space in the directories associated with a device class with a device type of FILE.

## Privilege class

Any administrator can issue this command.

## Syntax

```

>>—Query DIRSPace—┐
                    └─device_class_name─┘
  
```

## Parameters

*device\_class\_name*

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 of device type FILE are displayed. If you do not specify a value for this parameter, all device classes of device type FILE are displayed.

## Example: List FILE type device classes

Display information for all device classes with a device type of FILE. In the following example the unit M is equivalent to megabytes, and the unit G is equivalent to gigabytes.

query dirspace

| Device Class | Directory                  | Estimated Capacity | Estimated Available |
|--------------|----------------------------|--------------------|---------------------|
| DBBKUP       | /This/is/a/large/directory | 13,000 M           | 5,543 M             |
| DBBKUP       | /This/is/directory2        | 13,000 M           | 7,123 M             |
| DBBKUP2      | /This/is/a/huge/directory  | 2,256 G            | 2,200 G             |

## Related commands

Table 214. Commands related to QUERY DIRSPACE

| Command         | Description                                |
|-----------------|--|
| DEFINE DEVCLASS | Defines a device class.                    |
| DELETE DEVCLASS | Deletes a device class name.               |
| QUERY DEVCLASS  | Displays information about device classes. |
| UPDATE DEVCLASS | Changes the attributes of a device class.  |

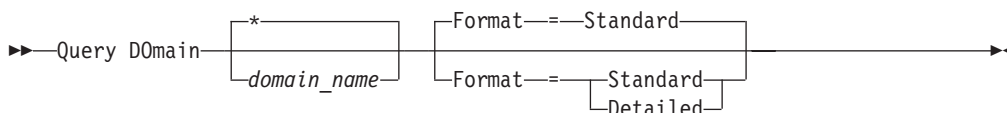
## QUERY DOMAIN (Query a policy domain)

Use this command to display information on one or more policy domains.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *domain\_name*

Specifies the policy domain to query. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a value for this parameter, all policy domains are displayed.

#### Format

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.

##### Detailed

Specifies that complete information is displayed.

### Example: Display a summary of policy domains

Display partial information for all policy domains on the server. Issue the command:

```
query domain
```

| Policy Domain Name | Activated Policy Set | Activated Default Mgmt Class | Number of Registered Nodes | Description                     |
|--------------------|----------------------|------------------------------|----------------------------|---------------------------------|
| EMPLOYEE-RECORDS   | VACATION             | ACTIVEFI-LES                 | 6                          | Employee Records Domain         |
| PROG1              |                      |                              | 0                          | Programming Group Test Domain   |
| PROG2              |                      |                              | 0                          | Programming Group Test Domain   |
| STANDARD           | STANDARD             | STANDARD                     | 1                          | Installed default policy domain |

See “Field descriptions” on page 639 for field descriptions.

### Example: Display the list of active-data pools

Display the active-data pool list. Issue the command:

```
query domain format=detailed
```

```

Policy Domain Name: STANDARD
Activated Policy Set: STANDARD
Activation Date/Time: 05/16/2006 16:18:05
Days Since Activation: 15
Activated Default Mgmt Class: STANDARD
Number of Registered Nodes: 1
Description: Installed default policy domain.
Backup Retention (Grace Period): 30
Archive Retention (Grace Period): 365
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 05/31/2006 15:17:48
Managing profile:
Changes Pending: Yes
Active Data Pool List: ADPPPOOL

```

See “Field descriptions” for field descriptions.

## Field descriptions

### Policy Domain Name

The name of the policy domain.

### Activated Policy Set

The name of the policy set that was last activated in the domain.

The definitions in the last activated policy set and the ACTIVE policy set are not necessarily identical. When you activate a policy set, the server copies the contents of the policy set to the policy set with the special name ACTIVE. The copied definitions in the ACTIVE policy set can be modified only by activating another policy set. You can modify the original policy set without affecting the ACTIVE policy set. Therefore, definitions in the policy set that was last activated might not be the same as those in the ACTIVE policy set.

### Activation Date/Time

The date and time that the policy set was activated.

### Days Since Activation

The number of days since the policy set was activated.

### Activated Default Mgmt Class

The assigned default management class for the policy set.

### Number of Registered Nodes

The number of client nodes registered to the policy domain.

### Description

The description of the policy domain.

### Backup Retention (Grace Period)

The number of days to retain inactive backup versions of files when any of the following conditions occur:

- A file is rebound to a new management class, 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)

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)

The administrator that defined or most recently updated the policy domain. If this field contains \$\$CONFIG\_MANAGER\$\$, the policy domain is associated with a profile that is managed by the configuration manager.

### Last Update Date/Time

When the administrator defined or most recently updated the policy domain.

### Managing Profile

The profile or profiles to which the managed server subscribed to get the definition of this policy domain.

### Changes Pending

Whether or not changes are being made but not activated. Once the changes are activated, the field resets to No.

### Active Data Pool List

The list of active-data pools in the domain.

## Related commands

*Table 215. Commands related to QUERY DOMAIN*

| Command       | Description   |
|---------------|---|
| COPY DOMAIN   | Creates a copy of a policy domain.  |
| DEFINE DOMAIN | Defines a policy domain that clients can be assigned to.                    |
| DELETE DOMAIN | Deletes a policy domain along with any policy objects in the policy domain. |
| UPDATE DOMAIN | Changes the attributes of a policy domain.                                  |

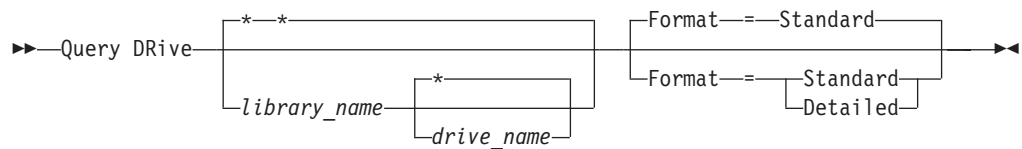
## QUERY DRIVE (Query information about a drive)

Use this command to display information about the drives associated with a library.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *library\_name*

Specifies the name of the library where the queried drive is located. This parameter is optional. You can use a wildcard character to specify this name.

You must specify a value for this parameter if you specify a drive name

#### *drive\_name*

Specifies the name assigned to the drive. This parameter is optional. You can use a wildcard character to specify this name. If you specify a drive name, you must also specify a *library\_name*.

### Format

Specifies how the information is displayed. This parameter is optional. The default 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.

### Example: List drives associated with the server

Display information about all drives associated with the server. Issue the command:

query drive

| Library Name | Drive Name | Device Type | Online |
|--------------|------------|-------------|--------|
| LIB1         | DRIVE01    | 3590        | Yes    |
| LIB2         | DRIVE02    | 3590        | Yes    |

See “Field descriptions” on page 642 for field descriptions.

### Example: Display detailed information on a specific drive and library

Display detailed information about the drive named DRIVE02 that is associated with the library LIB2. Issue the command:

## QUERY DRIVE

```
query drive lib2 drive02 format=detailed
```

```
Library Name: LIB2
Drive Name: DRIVE02
Device Type: 3590
On Line: Yes
Drive State: Empty
ACS DriveID:
Allocated to:
Last Update by (administrator): ADMIN
Last Update Date/Time: 02/29/2002 09:26:23
Cleaning Frequency (Gigabytes/ASNEEDED/NONE): NONE
```

See “Field descriptions” for field descriptions.

### Field descriptions

#### Library Name

The name of the library to which the drive is assigned.

#### Drive Name

The name assigned to the drive.

#### Device Type

The device type as specified in the associated device class. The server must have a path defined from the server to the drive in order to determine the true device type. As long as there is a path defined from the server to the drive, the server will display the true device type of the drive even if there are other paths defined to this drive. Exceptions to this occur if the device type is remote or unknown.

#### REMOTE

The server does not have a path to the device. The only defined paths to the device are from data movers.

#### UNKNOWN

No path exists.

**Tip:** Review the output of the QUERY PATH command to determine if the desired paths are defined. If they are not defined, define those desired paths using the DEFINE PATH command. Also, if using a data mover device, review the output of the QUERY DATAMOVER command to determine the type of the data mover device. If you are using a path from the server to a drive, the device type of the device class and the drive need to match. If you are using a path from a data mover device to a drive, review the documentation for your type of data mover to ensure the device type of the device class is compatible with the type of data mover device.

#### On Line

Specifies the status of the drive:

**Yes** The drive is online and available for server operations.

**No** The drive is offline and was put in this state by an administrator updating the status.

#### Unavailable Since

Specifies that the drive has been unavailable since *mm/dd/yy hh:mm:ss*. Output shows the time the server marked the drive as unavailable.



### Polling Since

Specifies that the server is polling the drive because the drive stopped responding. Output shows the time the server detected a problem and began polling. The server will poll a drive before stating it is unavailable. The time output follows the format: mm/dd/yy hh:mm:ss.

For more information on conditions under which the server will poll, see *Administrator's Guide*.

### Drive State

This specifies the current state of this particular drive based on the result of the last SCSI command to the drive or library. The server tracks the state of the drive to improve its selection of a drive for an operation and its drive recovery operations. The values are:

#### Unavailable

The drive is not available to the library for operations.

**Empty** The drive is empty and ready for operations.

#### Loaded

The drive is currently loaded, and the server is performing operations to the drive.

#### Unloaded

The media has been ejected from the drive.

#### Reserved

The drive is reserved for a mount request.

#### Unknown

The drive begins in drive state unknown as a result of being defined, as a result of server initialization, or as a result of having its status updated to online.

### ACS DriveId

The ID that specifies the physical location of the drive in an ACSLS library. This drive ID is specified as *a,l,p,d*, where *a* is the ACSID, *l* is the LSM (library storage module), *p* is the panel number, and *d* is the drive ID.

### Allocated To

The name of the library client that is currently using the drive. This applies to shared SCSI libraries only; the field is left blank for all other libraries.

### Last Update by (administrator)

Who performed the last update to the drive.

### Last Update Date/Time

The date and time when the last update occurred.

### Cleaning Frequency

How often the server activates drive cleaning. This value can be the number of gigabytes, NONE, or ASNEEDED.

## Related commands

Table 216. Commands related to QUERY DRIVE

| Command       | Description   |
|---------------|---|
| AUDIT LIBRARY | Ensures that an automated library is in a consistent state. |
| DEFINE DRIVE  | Assigns a drive to a library.                               |

## QUERY DRIVE

*Table 216. Commands related to QUERY DRIVE (continued)*

| Command        | Description                                       |
|----------------|---|
| DEFINE LIBRARY | Defines an automated or manual library.           |
| DEFINE PATH    | Defines a path from a source to a destination.    |
| DELETE DRIVE   | Deletes a drive from a library.                   |
| DELETE LIBRARY | Deletes a library.                                |
| QUERY LIBRARY  | Displays information about one or more libraries. |
| UPDATE DRIVE   | Changes the attributes of a drive.                |

## QUERY DRMEDIA (Query disaster recovery media)

Use this command to display information about database backup and copy storage pool volumes. You can also use the command to create a file of executable commands to process the volumes.

**Remember:** The QUERY DRMEDIA command always processes eligible copy storage-pool volumes. (For details about eligible copy storage pool volumes, see the description of the COPYSTGPOOL parameter in this command). By default, volumes in active-data pools are not eligible for processing by the disaster recovery manager. To process active-data pool volumes, you must issue the SET DRMACTIVEDATASTGPOOL command, or you must use the ACTIVATEDATASTGPOOL parameter on the QUERY DRMEDIA command. Depending on the setting for the SOURCE parameter in this command, database backups (full and incremental types, or snapshot types) can also be processed. The other parameters of the command (such as WHERESTATE) can further limit the results of the query.

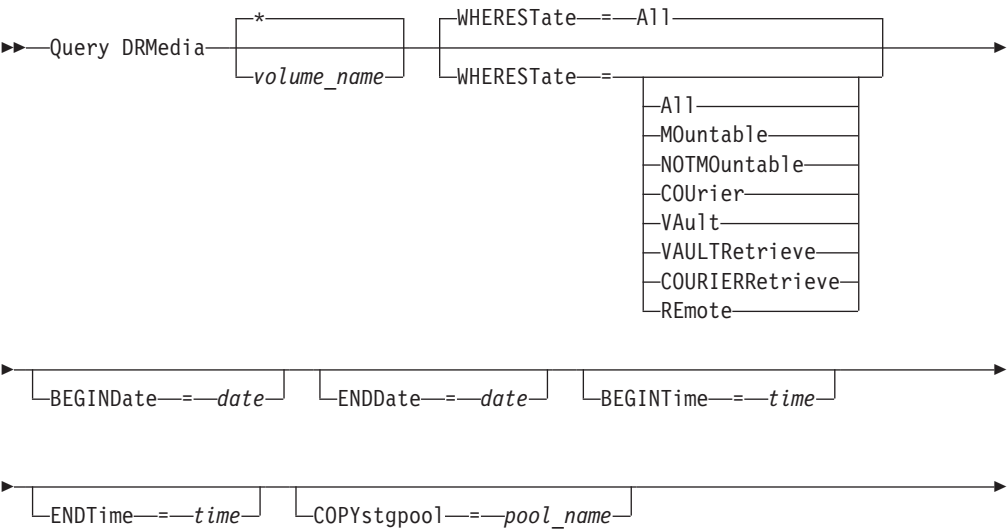
If you are using an external library and have moved a volume to the NOTMOUNTBLE state using the MOVE DRMEDIA command, the QUERY DRMEDIA command might still report the volume state as MOUNTABLE if it detects that the volume is in the library. Refer to the external library documentation for information about the procedures that you should follow when using the MOVE DRMEDIA and the QUERY DRMEDIA commands.

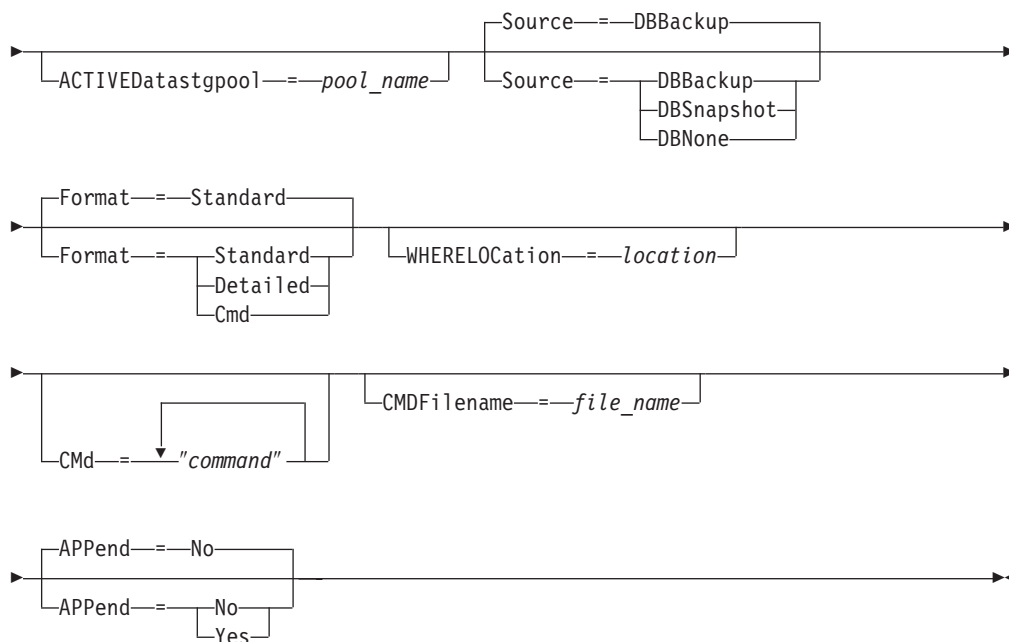
### Privilege class

To issue this command, you must have one of the following privilege classes:

- If the CMD parameter is NOT specified: operator or system privilege.
- If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO: operator, unrestricted storage, or system privilege.
- If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES (the default): system privilege.

### Syntax





## Parameters

### *volume\_name*

Specifies the names of the database backup and copy storage pool volumes to be queried. You can use wildcard characters to specify multiple names. This parameter is optional. The server looks for matching names among the following eligible volumes:

- Database backup volumes, as selected by the **SOURCE** parameter of this command.
- Copy storage pool volumes from copy storage pools named in the **COPYSTGPOOL** parameter. If you do not use the **COPYSTGPOOL** parameter, the server queries volumes from copy storage pools previously selected by the **SET DRMCOPYSTGPOOL** command.

Other parameters can also limit the results of the query.

### **WHEREState**

Specifies the state of volumes to be processed. This parameter is optional. The default is **ALL**. Possible values are:

#### **All**

Specifies all volumes in all states.

#### **MOuntable**

Volumes in this state contain valid data and are accessible for onsite processing.

#### **NOTMOuntable**

Volumes in this state are onsite, contain valid data, and not accessible for onsite processing.

#### **COUrier**

Volumes in this state are being moved to an offsite location.

#### **VAult**

Volumes in this state are offsite, contain valid data, and are not accessible for onsite processing.

### VAULTRetrieve

Volumes in this state are located at the offsite vault, do not contain valid data and can be moved back onsite for reuse or disposal:

- A copy storage pool volume is considered to be in the VAULTRETRIEVE state if it has been empty for at least the number of days specified with the REUSEDELAY parameter on the DEFINE STGPOOL command.
- A database backup volume is considered to be in the VAULTRETRIEVE state if it is associated with a database backup series that was expired based on the value specified using the SET DRMDBBACKUPEXPIREDAYS command.

**Important:** When you issue QUERY DRMEDIA WHERESTATE=VAULTRETRIEVE, the server dynamically determines which volumes can be moved back onsite for reuse or disposal. Therefore, to ensure that you identify all volumes that are in a VAULTRETRIEVE state, issue QUERY DRMEDIA WHERESTATE=VAULTRETRIEVE without the BEGINDATE, ENDDATE, BEGINTIME or ENDTIME parameters. The Last Update Date/Time field in the output for QUERY DRMEDIA WHERESTATE=VAULTRETRIEVE displays the date and time that a volume was moved to the VAULT state, not VAULTRETRIEVE.

### COURIERRetrieve

Volumes in this state are being moved back to the onsite location.

### REmote

Volumes in this state contain valid data and are located at the offsite remote server.

### BEGINDate

Specifies the beginning date used to select volumes. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or after the specified date. The default is the earliest date for which volume information exists.

You can specify the date using one of the following values:

| Value               | Description  | Example   |
|---------------------|--|---|
| MM/DD/YYYY          | A specific date  | 09/15/1998  |
| TODAY               | The current date   | TODAY   |
| TODAY-days or -days | The current date minus days specified. The maximum number of days is 9999. | TODAY"7 or "7.<br><br>To query volumes beginning with records changed to their current state a week ago, you can specify BEGINDATE=TODAY-7 or BEGINDATE=-7. |

### ENDDate

Specifies the ending date used to select volumes. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or before the specified date. The default is the current date.

You can specify the date using one of the following values:

| Value      | Description     | Example    |
|------------|-----------------|------------|
| MM/DD/YYYY | A specific date | 09/15/1998 |

| Value               | Description  | Example   |
|---------------------|--|---|
| TODAY               | The current date   | TODAY   |
| TODAY-days or -days | The current date minus days specified. The maximum number of days is 9999. | TODAY"7 or "7.<br><br>To query volumes beginning with records changed to their current state a week ago, you can specify<br>BEGINDATE=TODAY-7 or<br>BEGINDATE=-7. |

## BEGINTime

Specifies the beginning time used to select volumes. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or after the specified time and date. The default is midnight (00:00:00) on the date specified with the BEGINDATE parameter.

You can specify the time using one of the following values:

| Value               | Description  | Example   |
|---------------------|--|---|
| HH:MM:SS            | A specific time on the specified begin date                          | 12:33:28  |
| NOW                 | The current time on the specified begin date                         | NOW   |
| NOW+HH:MM or +HH:MM | The current time plus hours and minutes on the specified begin date  | NOW+03:00 or +03:00.<br><br>If you issue QUERY DRMEDIA command at 9:00 with<br>BEGINTIME=NOW+03:00 or<br>BEGINTIME=+03:00. Tivoli Storage Manager displays volumes that were changed to their current state at 12:00 on the begin date you specify. |
| NOW-HH:MM or -HH:MM | The current time minus hours and minutes on the specified begin date | NOW-03:30 or -03:30.<br><br>If you issue QUERY DRMEDIA command at 9:00 with<br>BEGINTIME=NOW-03:30 or<br>BEGINTIME=-03:30. Tivoli Storage Manager displays volumes that were changed to their current state at 5:30 on the begin date you specify.  |

## ENDTime

Specifies the ending time used to select volumes. This parameter is optional. Volumes are considered eligible if the MOVE DRMEDIA command has changed the volume to its current state on or before the specified time and date. The default is 23:59:59.

You can specify the time using one of the following values:

| Value    | Description                                | Example  |
|----------|--|----------|
| HH:MM:SS | A specific time on the specified end date  | 10:30:08 |
| NOW      | The current time on the specified end date | NOW      |

| Value                  | Description  | Example  |
|------------------------|--|--|
| NOW+HH:MM or<br>+HH:MM | The current time plus hours and minutes on the specified end date  | NOW+03:00 or +03:00.<br><br>If you issue QUERY DRMEDIA command at 9:00 with<br>ENDTIME=NOW+03:00 or<br>ENDTIME=+03:00, Tivoli Storage Manager processes volumes that were changed to their current state at 12:00 on the end date you specify. |
| NOW-HH:MM or<br>-HH:MM | The current time minus hours and minutes on the specified end date | NOW-03:30 or -03:30<br><br>If you issue QUERY DRMEDIA command at 9:00 with<br>ENDTIME=NOW-03:00 or<br>ENDTIME=-03:00, Tivoli Storage Manager processes volumes that were changed to their current state at 6:00 on the end date you specify.   |

### COPYstgpool

Specifies the name of the copy storage pool whose volumes are to be processed. This parameter is optional. You can use wildcard characters to specify this name. The copy storage pools specified with this parameter override those specified with the SET DRMCOPYSTGPOOL command.

If this parameter is not specified, Tivoli Storage Manager selects the storage pools as follows:

- If the SET DRMCOPYSTGPOOL command was previously issued with valid copy storage pool names, Tivoli Storage Manager processes only those storage pools.
- If the SET DRMCOPYSTGPOOL command has not been issued, or if all of the copy storage pools have been removed using the SET DRMCOPYSTGPOOL command, Tivoli Storage Manager processes all copy storage pool volumes in the specified state (ALL, MOUNTABLE, NOTMOUNTABLE, COURIER, VAULT, VAULTRETRIEVE, COURIERRETRIEVE, or REMOTE).

### Source

Specifies whether any database backup volumes are selected. This parameter is optional. The default is DBBACKUP. Possible values are:

#### DBBackup

Full and incremental database backup volumes are selected.

#### DBSnapshot

Snapshot database backup volumes are selected.

#### DBNone

No database backup volumes are selected.

### ACTIVEDatastgpool

Specifies the name of the active-data storage pool whose volumes are to be processed. This parameter is optional. You can use wildcard characters to specify this name. The active-data pools that are specified with this parameter override those specified with the SET DRMACTIVEDATASTGPOOL command.

If this parameter is not specified, Tivoli Storage Manager selects the storage pools as follows:

- If the SET ACTIVEDATASTGPOOL command was previously issued with valid active-data pool names, Tivoli Storage Manager processes only those storage pools.
- If the SET ACTIVEDATASTGPOOL command has not been issued, or all of the active-data pools have been removed using the SET ACTIVEDATASTGPOOL command, Tivoli Storage Manager processes all active-data pool volumes in the specified state (ALL, NOTMOUNTABLE, COURIER, VAULT, VAULTRETRIEVE, COURIERRETRIEVE, or REMOTE). Volumes in the MOUNTABLE state are not processed.

### Format

Specifies the information to be displayed. This parameter is optional. The default is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed.

#### Detailed

Specifies that detailed information is displayed.

#### Cmd

Specifies that executable commands are built for the selected volumes. If you specify FORMAT=CMD, you must also specify the CMD parameter.

### WHERELocation

Specifies the location of the volumes to be queried. This parameter is optional. The maximum length of the location is 255 characters. Enclose the text in quotation marks if it contains any blank characters. If you specify a target server name, the disaster recovery manager displays all database backup volumes and copy storage pool volumes located on the target server.

### CMd

Specifies the creation of executable commands to process the volume name and location obtained by this command. This parameter is optional. You must enclose the command specification in quotation marks. The maximum length of this parameter is 255 characters. The disaster recovery manager writes the commands to a file specified by the CMDFILENAME parameter or the SET DRMCMDFILENAME command, or generated by the QUERY DRMEDIA command. If the command length is greater than 240 characters, it is split into multiple lines and continuation characters (+) are added. You may need to alter the continuation character according to the product that runs the commands.

If you do not specify the FORMAT=CMD parameter, this command will not write create any command lines.

#### *string*

The command string. The string must not include embedded quotation marks. For example, this is a valid CMD parameter:

```
cmd="checkin libvol lib8mm &vol status=scratch"
```

This is an example of a CMD parameter that is *not* valid:

```
cmd=""checkin libvolume lib8mm" &vol status=scratch"
```

#### *substitution*

Specifies a substitution variable to tell QUERY DRMEDIA to substitute a value for the variable. The variables are not case-sensitive, and must not contain blank spaces after the ampersand (&). The possible variables are:

#### **&VOL**

A volume name variable.



### &LOC

A volume location.

### &VOLDSN

The name of the file the server writes into the sequential access media labels. An example of a copy storage pool tape volume file name using the default prefix TSM is TSM.BFS. An example of a database backup tape volume file name using a prefix TSM310 defined with the device class is TSM310.DBB.

### &NL

The new line character. When &NL is specified, QUERY DRMEDIA command splits the command at the &NL variable and does not append a continuation character. You must specify the proper continuation character before the &NL if one is required. If the &NL is not specified and the command line is greater than 240 characters, the line is split into multiple lines and continuation characters (+) are added.

### CMDFilename

Specifies the fully qualified name of the file to contain the commands specified with CMD parameter. This parameter is optional.

If you do not specify a name with the SET DRMCMDFILENAME command, Tivoli Storage Manager creates a file name by appending exec.cmds to the absolute directory path name of the Tivoli Storage Manager instance directory. If you specify a null string (" ") the commands are displayed on the console only. You can redirect the commands to a file using the redirection character for the operating system.

If the operation fails after the command file has been created, the file is not deleted.

### APPend

Specifies whether to overwrite any existing contents of the command file or append the commands to the file. This parameter is optional. The default is NO. Possible values are:

#### No

The disaster recovery manager overwrites the contents of the file.

#### Yes

The disaster recovery manager appends the commands to the file.

## Example: List volumes to be sent to offsite storage

Display all volumes to be given to a courier for offsite storage.

```
query drmedia wherestate=notmountable
format=standard
```

| Volume Name | State         | Last Update<br>Date/Time | Automated<br>LibName |
|-------------|---------------|--------------------------|----------------------|
| TAPE01      | Not mountable | 01/20/1998 14:25:22      |                      |
| DBTP01      | Not mountable | 01/20/1998 14:25:22      |                      |
| DBTP03      | Not mountable | 01/20/1998 14:31:53      |                      |

See "Field descriptions" on page 652 for field descriptions.

## Example: Display information on volumes at the vault

Display detailed information about all volumes at the vault.

```
query drmedia wherestate=vault format=detailed
```

```

Volume Name: DBTP02
State: Vault
Last Update Date/Time: 01/20/1998 13:29:02
Location: Ironmnt
Volume Type: DBBackup
Copy Storage Pool Name:
Active-Data Storage Pool Name: TSMACTIVEPOOL
Automated LibName:
```

See “Field descriptions” for field descriptions.

## Field descriptions

### Volume Name

The name of the database backup or copy storage pool volume.

**State** The state of the volume.

### Last Update Date/Time

The date and time that the volume state was last updated. For volumes in the VAULTRETRIEVE state, this field displays the date and time that a volume was moved to the VAULT state, not VAULTRETRIEVE. The server does not “update” volumes to VAULTRETRIEVE. At the time the QUERY DRMEDIA command is issued, the server dynamically determines whether the data in copy storage pool volumes and database backup volumes is no longer valid and whether the volume can be brought back onsite for reuse or disposal.

### Location

The volume location.

### Volume Type

The type of volume. Possible values are:

#### DBBackup

A full or incremental database backup volume.

#### DBSnapshot

A database snapshot backup volume.

#### CopyStgPool

A copy storage pool volume.

### Copy Storage Pool Name

For a copy storage pool volume, the name of the copy storage pool.

### Active-Data Storage Pool Name

For an active-data pool volume, the name of the active-data pool.

### Automated LibName

The name of the automated library if the volume is in a library.

## Related commands

Table 217. Commands related to QUERY DRMEDIA

| Command                   | Description  |
|---------------------------|--|
| BACKUP DB                 | Backs up the IBM Tivoli Storage Manager database to sequential access volumes.                                     |
| BACKUP STGPOOL            | Backs up a primary storage pool to a copy storage pool.  |
| CHECKOUT LIBVOLUME        | Checks a storage volume out of an automated library.   |
| MOVE DRMEDIA              | Moves DRM media on-site and off-site.  |
| SET DRMACTIVEDATASTGPOOL  | Specifies that active-data storage pools are managed by DRM.   |
| SET DRMCOPYSTGPOOL        | Specifies that copy storage pools are managed by DRM.  |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.  |
| SET DRMCMDFILENAME        | Specifies a file name for containing DRM executable commands.  |
| SET DRMFILEPROCESS        | Specifies whether the MOVE DRMEDIA or QUERY DRMEDIA command processes files associated with a device type of file. |

### QUERY DRMSTATUS (Query disaster recovery manager system parameters)

Use this command to display information about the system parameters defined for disaster recovery manager (DRM).

#### Privilege class

Any administrator can issue this command.

#### Syntax

►►—Query DRMStatus—◄◄

#### Parameters

None.

#### Example: Display DRM system parameter information

Display information about the DRM system parameters:

```
query drmstatus
```

```
Recovery Plan Prefix:
Plan Instructions Prefix:
Replacement Volume Postfix: @
Primary Storage Pools: PRIM1 PRIM2
Copy Storage Pools: COPY*
Active-Data Storage Pools: TSMACTIVEPOOL
Not Mountable Location name: Local
Courier Name: Fedex
Vault Site Name: Ironmnt
DB Backup Series expiration days: 30 Day(s)
Recovery Plan File Expiration Days: 30 Days(s)
Check Label?: No
Process FILE Device Type?: No
Command file name:
```

#### Field descriptions

##### Recovery Plan Prefix

User-specified prefix portion of the file name for the recovery plan file.

##### Plan Instructions Prefix

User-specified prefix portion of the file names for the server recovery instructions files.

##### Replacement Volume Postfix

The character added to the end of the replacement volume names in the recovery plan file.

##### Primary Storage Pools

The primary storage pools that are eligible for processing by the PREPARE command. If this field is blank, all primary storage pools are eligible.

##### Copy Storage Pools

The copy storage pools that are eligible for processing by the MOVE DRMEDIA, PREPARE, and QUERY DRMEDIA commands. If this field is blank, all copy storage pools are eligible.

**Active-data Storage Pools**

The active-data pools that are eligible for processing by the MOVE DRMEDIA, PREPARE, and QUERY DRMEDIA commands. If this field is blank, active-data pools are not eligible.

**Not Mountable Location Name**

The name of the offsite location where the media to be shipped are stored.

**Courier Name**

The name of the courier used to carry the media to the vault.

**Vault Site Name**

The name of the vault where the media is stored.

**DB Backup Series Expiration Days**

The minimum number of days that must elapse since a database series has been created before it is eligible to be expired. See the SET DRMDBBACKUPEXPIREDAYS command for information about the criteria that must be met for database backup series expiration.

**Recovery Plan File Expiration Days**

The minimum number of days that must elapse since a recovery plan file, stored on a target server, has been created before it is eligible to be expired. See the SET DRMRPFEXPIREDAYS command for information about the criteria that must be met for recovery plan file expiration.

**Check Label?**

Whether media labels are read for sequential media volumes checked out by the MOVE DRMEDIA command. Possible values are Yes or No.

**Process FILE Device Type?**

Whether MOVE DRMEDIA or QUERY DRMEDIA commands process database backup and copy storage pool volumes associated with a device class with a FILE device type. Possible values are Yes or No.

**Command File Name**

The full path file name that contains the executable commands generated by the MOVE DRMEDIA or QUERY DRMEDIA command.

**Related commands**

*Table 218. Commands related to QUERY DRMSTATUS*

| Command                   | Description  |
|---------------------------|--|
| SET DRMCHECKLABEL         | Specifies whether IBM Tivoli Storage Manager should read volume labels during MOVE DRMEDIA command processing. |
| SET DRMACTIVEDATASTGPOOL  | Specifies that active-data storage pools are managed by DRM.   |
| SET DRMCOPYSTGPOOL        | Specifies that copy storage pools are managed by DRM.  |
| SET DRMCMDFILENAME        | Specifies a file name for containing DRM executable commands.  |
| SET DRMCOURIERNAME        | Specifies the name of the courier for the disaster recovery media.   |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.  |

## QUERY DRMSTATUS

*Table 218. Commands related to QUERY DRMSTATUS (continued)*

| Command                 | Description  |
|-------------------------|--|
| SET DRMFILEPROCESS      | Specifies whether the MOVE DRMEDIA or QUERY DRMEDIA command processes files associated with a device type of file. |
| SET DRMINSTRPREFIX      | Specifies the prefix portion of the path name for the recovery plan instructions.                                  |
| SET DRMPLANVPOSTFIX     | Specifies the replacement volume names in the recovery plan file.  |
| SET DRMPLANPREFIX       | Specifies the prefix portion of the path name for the recovery plan.   |
| SET DRMPRIMSTGPOOL      | Specifies that primary storage pools are managed by DRM.   |
| SET DRMRPFEXPIREDAYS    | Set criteria for recovery plan file expiration.  |
| SET DRMVAULTNAME        | Specifies the name of the vault where DRM media is stored.   |
| SET DRMNOTMOUNTABLENAME | Specifies the location name of the DRM media to be sent offsite.   |

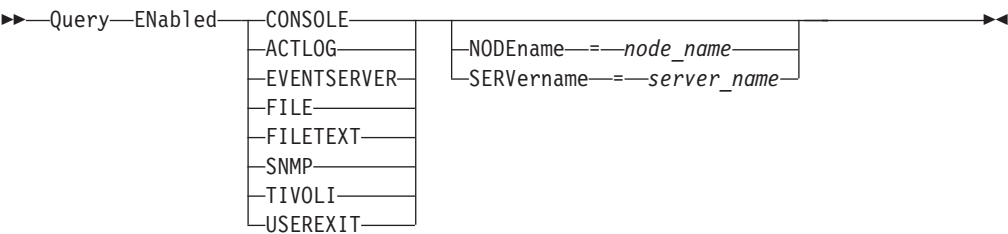
## QUERY ENABLED (Query enabled events)

Use this command to display either a list of enabled events or a list of disabled events, whichever is shorter.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

*receiver*

Specifies a type of receiver for enabled events. This is a required parameter. Valid values are:

#### ACTLOG

Specifies the Tivoli Storage Manager activity log as a receiver.

#### CONSOLE

Specifies the standard server console as a receiver.

#### EVENTSERVER

Specifies the event server as a receiver.

#### FILE

Specifies a user file as a receiver. Each logged event is a record in the file and a person cannot read each logged event easily.

#### FILETEXT

Specifies a user file as a receiver. Each logged event is a fixed-size, readable line.

#### SNMP

Specifies the simple network management protocol (SNMP) as a receiver.

#### TIVOLI

Specifies the Tivoli Management Environment (TME) as a receiver.

#### USEREXIT

Specifies a user-written routine to which Tivoli Storage Manager writes information as a receiver.

#### NODEname

Specifies a node name to be queried. You can specify NODENAME or SERVERNAME. If neither parameter is specified, the query is for events enabled for the server running this command.

#### SERVername

Specifies a server name to be queried. You can specify NODENAME or

## QUERY ENABLED

SERVERNAME. If neither parameter is specified, the query is for events enabled for the server running this command.

### Example: Query the server for console events

Query the server for server events that are enabled for the console. There are 10000 possible server events. Either a list of enabled events or disabled events is displayed (whichever list is shorter).

```
query enabled console
```

9998 events are enabled for the CONSOLE receiver. The following events are DISABLED for the CONSOLE receiver:

ANR8409, ANR8410

### Related commands

*Table 219. Commands related to QUERY ENABLED*

| Command            | Description   |
|--------------------|---|
| BEGIN EVENTLOGGING | Starts event logging to a specified receiver.   |
| DISABLE EVENTS     | Disables specific events for receivers.   |
| ENABLE EVENTS      | Enables specific events for receivers.  |
| END EVENTLOGGING   | Ends event logging to a specified receiver.   |
| QUERY EVENTRULES   | Displays information about rules for server and client events.                          |
| QUERY STATUS       | Displays the settings of server parameters, such as those selected by the SET commands. |



## QUERY EVENT (Query scheduled and completed events)

Use this command to display the status of scheduled events. The time and date parameters allow you to limit the query to events that were scheduled to occur within the specified times and dates. Limiting the output to events whose scheduled start times fall within a date and time range also minimizes the time it takes to process this query.

The command syntax differs for queries that apply to scheduled client operations and to scheduled administrative commands.

*Table 220. Commands related to QUERY EVENT*

| Command            | Description  |
|--------------------|--|
| DEFINE SCHEDULE    | Defines a schedule for a client operation or an administrative command.                          |
| DELETE EVENT       | Deletes event records prior to a specified date and time.  |
| QUERY ACTLOG       | Displays messages from the server activity log.  |
| SET EVENTRETENTION | Specifies the number of days to retain records for scheduled operations.                         |
| SET RANDOMIZE      | Specifies the randomization of start times within a window for schedules in client-polling mode. |

## QUERY EVENT – client schedules

### QUERY EVENT (Display client schedules)

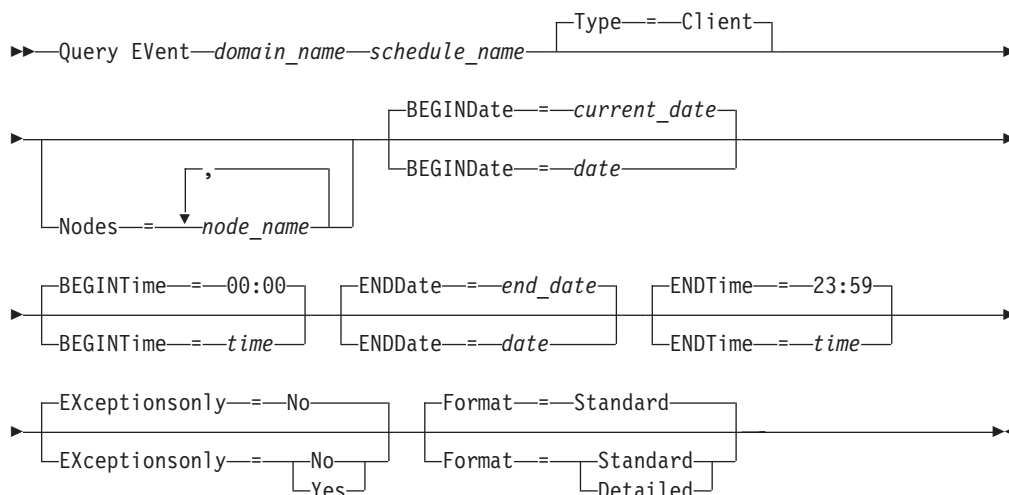
Use the QUERY EVENT command to display scheduled and completed events for selected clients.

Tivoli Storage Manager keeps only one version of an event record in the database. If a client schedule is changed, all previous event records for that schedule are removed from the database.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domain\_name* (Required)

Specifies the name of the policy domain to which the schedules belong. You can use a wildcard character to specify this name.

##### *schedule\_name* (Required)

Specifies the name of the schedule for which events are displayed. You can use a wildcard character to specify this name.

##### Type=Client

Specifies that the query displays events for client schedules. This parameter is optional. The default is CLIENT.

##### Nodes

Specifies the name of the client node that belongs to the specified policy domain for which events are displayed. You can specify multiple client nodes by separating the names with commas and no intervening spaces. You can use wildcard characters to specify nodes. If you do not specify a client name, events display for all clients that match the domain name and the schedule name.

##### BEGINDate

Specifies the beginning date of the time range for events to be displayed. All events scheduled to start during this time are displayed. This parameter is optional. The default is the current date.

You can specify the date using one of the values below:

| Value                                | Description   | Example  |
|--------------------------------------|---|--|
| <i>MM/DD/YYYY</i>                    | A specific date   | 09/15/1998   |
| <b>TODAY</b>                         | The current date  | TODAY  |
| <b>TODAY+days</b> or<br><i>+days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3.  |
| <b>TODAY-days</b> or<br><i>-days</i> | The current date minus days specified   | TODAY-7 or -7.<br><br>To query events scheduled to start during the past seven days, specify<br>BEGINDATE=TODAY-7<br>ENDDATE=TODAY or BEGINDATE=-7<br>ENDDATE=TODAY. |

### BEGINTime

Specifies the beginning time of the range for events to be displayed. All events scheduled to start during this time are displayed. This parameter is optional. The default value is 00:00.

You can specify the time using one of the values below:

| Value                                | Description  | Example   |
|--------------------------------------|--|---|
| <i>HH:MM:SS</i>                      | A specific time on the specified begin date                          | 10:30:08  |
| <b>NOW</b>                           | The current time on the specified begin date                         | NOW   |
| <b>NOW+HH:MM</b><br>or <b>+HH:MM</b> | The current time plus hours and minutes on the specified begin date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 to query events scheduled to start 3 hours from now, you can specify either<br>BEGINTIME=NOW+03:00 or<br>BEGINTIME=+03:00. Tivoli Storage Manager displays events at 12:00 on the specified begin date.                                     |
| <b>NOW-HH:MM</b><br>or <b>-HH:MM</b> | The current time minus hours and minutes on the specified begin date | NOW-04:00 or -04:00.<br><br>If you issue this command at 9:00 to query events scheduled to start during the last 4 hours, you can specify either<br>BEGINTIME=NOW-04:00<br>ENDTIME=NOW or<br>BEGINTIME=-04:00 ENDTIME=NOW.<br>Tivoli Storage Manager displays events at 5:00 on the specified begin date. |

### ENDDate

Specifies the ending date of the time range for events to be displayed. All events that were schedule to start during this time are displayed. This parameter is optional. The default is the value used for the BEGINDATE.

## QUERY EVENT – client schedules

You can specify the date using one of the values below:

| Value                                | Description   | Example   |
|--------------------------------------|---|---|
| <i>MM/DD/YYYY</i>                    | A specific date   | 09/15/1998  |
| <b>TODAY</b>                         | The current date  | TODAY   |
| <b>TODAY+days</b> or<br><i>+days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3.   |
| <b>TODAY-days</b> or<br><i>-days</i> | The current date minus days specified   | TODAY-8 or -8.<br><br>To query events scheduled to start during a one-week period that ended yesterday, you can specify either<br>BEGINDATE=TODAY-8<br>ENDDATE=TODAY-1 or<br>BEGINDATE=-8 ENDDATE=-1. |

### ENDTime

Specifies the ending time of the 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 23:59.

You can specify the time using one of the values below:

| Value                                | Description  | Example   |
|--------------------------------------|--|---|
| <i>HH:MM:SS</i>                      | A specific time on the specified end date                          | 10:30:08  |
| <b>NOW</b>                           | The current time on the specified end date                         | NOW   |
| <b>NOW+HH:MM</b><br>or <b>+HH:MM</b> | The current time plus hours and minutes on the specified end date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 to query events scheduled to start 3 hours from now, you can specify either<br>BEGINTIME=NOW<br>ENDTIME=NOW+03:00 or<br>BEGINTIME=NOW ENDTIME=+03:00. |
| <b>NOW-HH:MM</b><br>or <b>-HH:MM</b> | The current time minus hours and minutes on the specified end date | NOW-04:00 or -04:00   |

### EXceptiononly

Specifies the type of information you want on scheduled or completed events. This parameter is optional. The default is NO. Possible values are:

#### No

Specifies that the information on past and projected events is displayed.

#### Yes

Specifies that the events that failed or did not process as scheduled are displayed.

### Format

Specifies how information displays. This parameter is optional. The default is STANDARD. Possible values are:

### Standard

Specifies that partial information for events displays.

### Detailed

Specifies that complete information for events displays.

### Example: Display partial information for unsuccessful events

Display partial information for all events scheduled for DOMAIN1 that did not run successfully. Limit the search to the client named JOE. Limit the events displayed to those that were scheduled to occur from February 11, 2001 (02/11/2001) to February 12, 2001 (02/12/2001).

```
query event domain1 * nodes=joe begindate=02/11/2001
enddate=02/12/2001 exceptionsonly=yes
```

| Scheduled Start     | Actual Start        | Schedule Name | Node Name | Status |
|---------------------|---------------------|---------------|-----------|--------|
| 02/11/1999 01:00:00 | 02/11/1999 01:13:55 | BACK1         | JOE       | Failed |
| 02/12/1999 01:00:00 |                     | DAILYBKP      | JOE       | Missed |

See “Field descriptions” on page 664 for field descriptions.

### Example: Display partial information for scheduled events for a client

Display complete information for all events scheduled for processing by client JOE in DOMAIN1. Use any time from January 15, 1999 (01/15/1999) through January 20, 1999 (01/20/1999).

```
query event domain1 * nodes=joe
begindate=01/15/1999 enddate=01/20/1999 format=detailed
```

```
Policy Domain Name: DOMAIN1
Schedule Name: DAILY_BACKUP
Node Name: JOE
Scheduled Start: 01/16/1999 01:00:00
Actual Start:
Completed:
Status: Missed
Result: 0
Reason:

Policy Domain Name: DOMAIN1
Schedule Name: DAILY_BACKUP
Node Name: JOE
Scheduled Start: 01/16/1999 01:00:00
Actual Start: 01/16/1999 01:13:55
Completed: 01/16/1999 01:18:27
Status: Completed
Result: 4
Reason: All operations completed successfully,
but some files were not processed

Policy Domain Name: DOMAIN1
Schedule Name: DAILY_BACKUP
Node Name: JOE
Scheduled Start: 01/16/1999 01:00:00
Actual Start: 01/16/1999 01:13:55
Completed: 01/16/1999 01:18:27
Status: Completed
Result: 8
Reason: All operations completed, with at least one
warning message
```

See “Field descriptions” on page 664 for field descriptions.

### Example: Display detailed information for scheduled events for a client

Display the detailed information for events scheduled for processing by client DOC between the hours of 10:00 a.m. and 11:00 a.m. on November 1, 2005 (11/01/2005). Notice that when the status is FAILED, the result code is displayed.

```
query event domain1 * nodes=doc begindate=11/01/2005
beginime=10:00 endtime=11:00 enddate=11/01/2005
exceptiononly=yes format=detailed
```

| Scheduled Start     | Actual Start        | Schedule Name | Node Name | Status    |
|---------------------|---------------------|---------------|-----------|-----------|
| 11/01/2005 10:01:01 | 11/01/2005 10:03:46 | T1            | DOC       | Failed 8  |
| 11/01/2005 10:16:01 | 11/01/2005 10:16:10 | T1            | DOC       | Failed 4  |
| 11/01/2005 10:31:01 | 11/01/2005 10:33:08 | T1            | DOC       | Completed |
| 11/01/2005 10:46:01 |                     | T1            | DOC       | Missed    |
| 11/01/2005 10:57:49 | 11/01/2005 10:58:07 | T0            | DOC       | Failed 12 |

### Field descriptions

#### Policy Domain Name

Specifies the name of the policy domain to which the schedule is assigned.

#### Schedule Name

Specifies the name of the schedule that 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.

#### Actual Start

Specifies the date and time at which the client began processing the scheduled operation. No information is displayed if the scheduled operation has not started.

#### Completed

Specifies the date and time the scheduled event is completed.

**Status** Specifies the status of the event at the time the QUERY EVENT command is issued. Possible values are:

#### Completed

Specifies that the scheduled event has been completed.

**Failed** Specifies that the client reports a failure running the scheduled operation and successive retries have not succeeded.

#### Failed - no restart

Specifies an intermediate status, when a client session is interrupted by a communications error or timeout on the server. This status can be changed to a final status of "Completed" or "Failed" when the event completes.

**Future** Specifies that the beginning of the startup window for the event is in the future. This status also indicates that an event record has not been created for this event.

#### In Progress

Specifies that the scheduled event is running and has not yet reported the completion state to the server.

Periodically check the status for completion of the scheduled event. If this status is not updated in a reasonable amount of time, review your client dsm Sched.log and dsmerror.log to determine why the client did not report the outcome of this event to the server. If the scheduled backup failed, rerun the scheduled event or perform a manual incremental backup to ensure the data backup.

### Missed

Specifies that the scheduled startup window for this event has passed and the schedule has not begun.

### Pending

Specifies that the QUERY EVENT command was issued during the startup window for the event, but processing the scheduled operation has not yet begun.

### Restarted

Specifies that the client has retried processing the scheduled operation.

### Severed

Specifies that the communications with the client is severed before the event can complete.

### Started

Specifies that the event has begun processing.

### Uncertain

Specifies that the state of the event cannot be determined. The server specifies Uncertain 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 do not reside in the database. If you do not want these records to display, either specify EXCEPTIONSONLY=YES or delete the schedule if it is no longer needed.

**Attention:** When a scheduled operation is processing, without having been restarted within its specified duration, the Status field shows Started. If the operation continues beyond the specified duration, no event record is created. If a query is issued after the specified duration has passed, the Status shows as Failed even if the operation is still running. After the operation completes, an event record is created, and a subsequent query shows the result in the Status field.

**Result** Specifies the return code from the client that identifies whether the schedule has processed successfully. If the return code is other than a zero, examine the client's error log and schedule log.

| Return code | Explanation  |
|-------------|--|
| 0           | All operations completed successfully.   |
| 4           | The operation completed successfully, but some files were not processed.                               |
| 8           | The operation completed with at least one warning message.   |
| 12          | The operation completed with at least one error message (except for error messages for skipped files). |

## QUERY EVENT – client schedules

If a schedule has ACTION=COMMAND as a parameter, and the command is not a Tivoli Storage Manager command, the command can produce other values in the Result field. See the *Backup-Archive Clients Installation and User's Guide* for details.

### **Reason**

Specifies the reason for the return code.



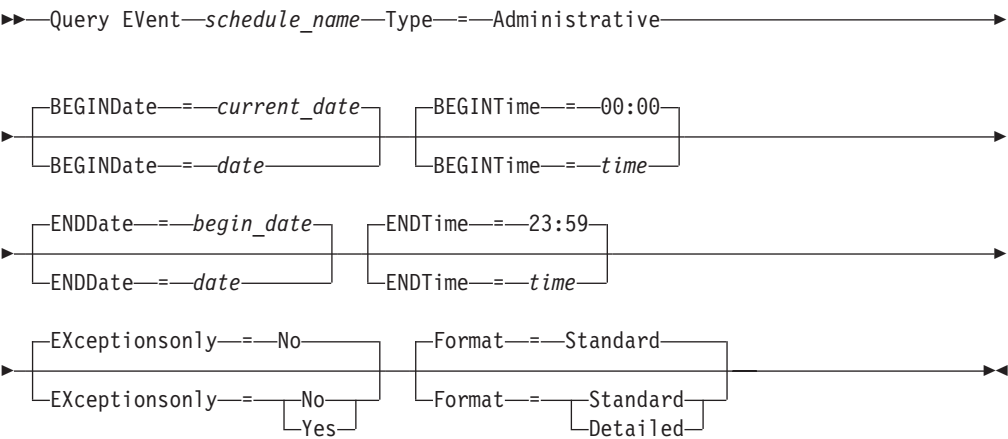
### QUERY EVENT (Display administrative event schedules)

Use the QUERY EVENT command to display scheduled and completed events for selected administrative command schedules.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *schedule\_name* (Required)

Specifies the name of the schedule for which events display. You can use wildcard characters to specify names.

##### Type=Administrative (Required)

Specifies that the query displays events for administrative command schedules.

##### BEGINDate

Specifies the beginning date of the time range for events to be displayed. All events scheduled to start during this time are displayed. This parameter is optional. The default is the current date.

You can specify the date using one of the values below:

| Value                             | Description   | Example  |
|-----------------------------------|---|--|
| <i>MM/DD/YYYY</i>                 | A specific date   | 09/15/1998   |
| <b>TODAY</b>                      | The current date  | TODAY  |
| <b>TODAY+days</b> or <b>+days</b> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3.  |
| <b>TODAY-days</b> or <b>-days</b> | The current date minus days specified   | TODAY-7 or -7.<br><br>To query events scheduled to start during the past seven days, specify<br>BEGINDATE=TODAY-7<br>ENDDATE=TODAY or BEGINDATE=-7<br>ENDDATE=TODAY. |

## QUERY EVENT — administrative schedules

### BEGINTime

Specifies the beginning time of the range for events to be displayed. All events scheduled to start during this time are displayed. This parameter is optional. The default value is 00:00.

You can specify the time using one of the values below:

| Value                                | Description  | Example  |
|--------------------------------------|--|--|
| <i>HH:MM:SS</i>                      | A specific time on the specified begin date                          | 10:30:08   |
| <b>NOW</b>                           | The current time on the specified begin date                         | NOW  |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes on the specified begin date  | NOW+03:00 <b>or</b> +03:00.<br><br>If you issue this command at 9:00 to query events scheduled to start 3 hours from now, you can specify either BEGINTIME=NOW+03:00 or BEGINTIME=+03:00. Tivoli Storage Manager displays events at 12:00 on the specified begin date.                                     |
| <b>NOW-HH:MM</b><br><b>or -HH:MM</b> | The current time minus hours and minutes on the specified begin date | NOW-04:00 <b>or</b> -04:00.<br><br>If you issue this command at 9:00 to query events scheduled to start during the last 4 hours, you can specify either BEGINTIME=NOW-04:00<br>ENDTIME=NOW or<br>BEGINTIME=-04:00 ENDTIME=NOW. Tivoli Storage Manager displays events at 5:00 on the specified begin date. |

### ENDDate

Specifies the ending date of the time range for events to be displayed. All events that were schedule to start during this time are displayed. This parameter is optional. The default is the value used for the BEGINDATE.

You can specify the date using one of the values below:

| Value                                       | Description   | Example   |
|---|---|---|
| <i>MM/DD/YYYY</i>                           | A specific date   | 09/15/1998  |
| <b>TODAY</b>                                | The current date  | TODAY   |
| <b>TODAY+days</b> <b>or</b><br><i>+days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 <b>or</b> +3.  |
| <b>TODAY-days</b> <b>or</b><br><i>-days</i> | The current date minus days specified   | TODAY-8 <b>or</b> -8.<br><br>To query events scheduled to start during a one-week period that ended yesterday, you can specify either BEGINDATE=TODAY-8<br>ENDDATE=TODAY-1 or<br>BEGINDATE=-8 ENDDATE=-1. |

### ENDTime

Specifies the ending time of the 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 23:59.

You can specify the time using one of the values below:

| Value                                | Description  | Example   |
|--------------------------------------|--|---|
| <i>HH:MM:SS</i>                      | A specific time on the specified end date                          | 10:30:08  |
| <b>NOW</b>                           | The current time on the specified end date                         | NOW   |
| <b>NOW+HH:MM</b><br><b>or +HH:MM</b> | The current time plus hours and minutes on the specified end date  | NOW+03:00 <b>or</b> +03:00.<br><br>If you issue this command at 9:00 to query events scheduled to start 3 hours from now, you can specify either<br>BEGINTIME=NOW<br>ENDTIME=NOW+03:00 <b>or</b><br>BEGINTIME=NOW ENDTIME=+03:00. |
| <b>NOW-HH:MM</b><br><b>or -HH:MM</b> | The current time minus hours and minutes on the specified end date | NOW-04:00 <b>or</b> -04:00  |

### EXceptiononly

Specifies the type of information you want on scheduled or completed events. This parameter is optional. The default is NO. Possible values are:

#### No

Specifies that the information on past and projected events is displayed.

#### Yes

Specifies that the events that failed or did not process as scheduled are displayed.

### Format

Specifies how the information displays. This parameter is optional. The default value is STANDARD. Possible values are:

#### Standard

Specifies that partial information for events displays.

#### Detailed

Specifies that complete information for events displays.

### Example: List events for a specific administrative schedule

Display partial information for all events scheduled for an administrative schedule named DOSADMIN. Limit the query to events that are scheduled for March 30, 1999 (03/30/1999). Issue the command:

```
query event dosadmin type=administrative
begindate=03/30/1999
enddate=03/30/1999
```

## QUERY EVENT — administrative schedules

| Scheduled Start     | Actual Start        | Schedule Status |           |
|---------------------|---------------------|-----------------|-----------|
|                     |                     | Name            |           |
| 03/30/1999 00:00:00 | 03/30/1999 00:00:01 | DOSADMIN        | Completed |
| 03/30/1999 04:00:00 | 03/30/1999 04:00:01 | DOSADMIN        | Completed |
| 03/30/1999 12:00:00 |                     | DOSADMIN        | Future    |
| 03/30/1999 16:00:00 |                     | DOSADMIN        | Future    |

### Field descriptions

#### Scheduled Start

Specifies the scheduled starting date and time for the event.

#### Actual Start

Specifies the date and time at which the client began processing the scheduled operation. No information displays if the schedule has not started executing.

#### Schedule Name

Specifies the name of the schedule that initiated this event.

**Status** For administrative commands or scripts that specify WAIT=YES, the status of a scheduled event is STARTED until the operation specified by the command or script is completed. The final status of the scheduled event depends on the return code of the operation. However, if WAIT=YES and if the schedule is running a script that specifies PREVIEW=YES, the final status is COMPLETED, unless the script contained a syntax error.

For administrative commands or scripts that specify WAIT=NO, the status of a scheduled event is COMPLETED if the scheduled command or script started. The success of the schedule is independent of the success of the operation performed by the command or script.

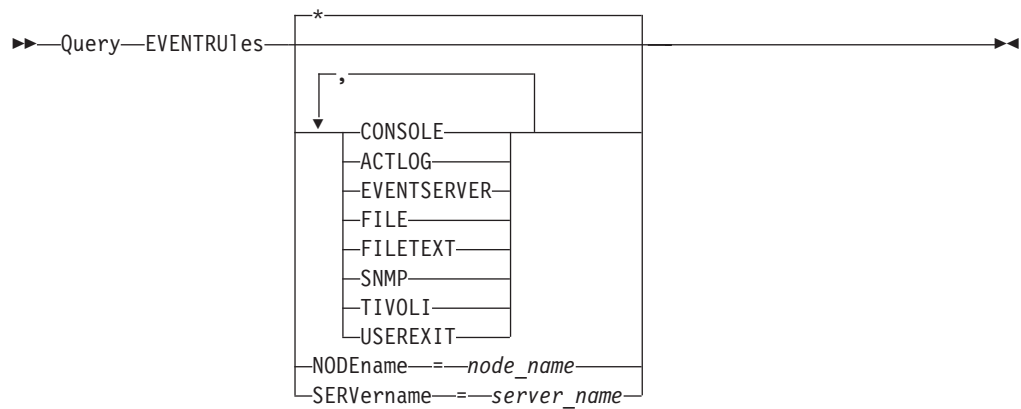
### QUERY EVENTRULES (Query rules for server or client events)

Use this command to display the history of events that are enabled or disabled by a specified receiver for the server or for a client node.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

*receivers*

Specifies the name of one or more receivers for enabled events. This parameter is optional.

You can use a wildcard character to specify all receivers.

Valid values are:

## CONSOLE

Specifies the standard console as a receiver.

## ACTLOG

Specifies the Tivoli Storage Manager activity log as a receiver.

## EVENTSERVER

Specifies the event server as a receiver.

## FILE

Specifies a user file as a receiver. Each logged event is a record in the file and a person cannot read each logged event easily.

## FILETEXT

Specifies a user file as a receiver. Each logged event is a fixed-size, readable line.

## SNMP

Specifies the simple network management protocol (SNMP) as a receiver.

## TIVOLI

Specifies the Tivoli Management Environment (TME) as a receiver.

## USEREXIT

Specifies a user-written routine to which Tivoli Storage Manager writes information as a receiver.

## QUERY EVENTRULES

### NODENAME

Specifies a node name to be queried. You can use a wildcard character to specify a name. You can specify NODENAME or SERVERNAME. If neither parameter is specified, the query is for event rules for the server running this command.

### SERVER

Specifies a server name to be queried. You can use a wildcard character to specify a name. You can specify NODENAME or SERVERNAME. If neither parameter is specified, the query is for event rules for the server running this command.

### Example: Display the history of client events for the server console

Display the history of client events enabled or disabled for the server console and activity log receivers.

```
query eventrules console,actlog nodename=*
```

| Date/Time         | Client Event Rules                            |
|-------------------|---|
| 05/29/97 13:39:58 | ENABLE EVENTS CONSOLE ANE4001 NODENAMES=JEE   |
| 05/30/97 13:46:25 | DISABLE EVENTS ACTLOG ANE4962 NODENAMES=JEE   |
| 05/30/97 13:46:25 | DISABLE EVENTS ACTLOG ANE4963 NODENAMES=JEE   |
| 05/30/97 13:46:25 | DISABLE EVENTS ACTLOG ANE4965 NODENAMES=JEE   |
| 05/30/97 13:46:25 | DISABLE EVENTS ACTLOG ANE4966 NODENAMES=JEE   |
| 05/30/97 13:46:25 | DISABLE EVENTS ACTLOG ANE4967 NODENAMES=JEE   |
| 05/30/97 13:46:25 | DISABLE EVENTS ACTLOG ANE4968 NODENAMES=JEE   |
| 05/30/97 14:24:20 | ENABLE EVENTS CONSOLE ANE4015 NODENAMES=RON   |
| 05/30/97 14:24:50 | ENABLE EVENTS CONSOLE ANE4026 NODENAMES=DONNA |
| 05/30/97 14:25:59 | ENABLE EVENTS CONSOLE ANE4015 NODENAMES=DONNA |

### Example: Display the history of client events for all receivers

Display the history of server events enabled or disabled for all receivers.

```
query eventrules
```

| Date/Time         | Server Event Rules             |
|-------------------|--------------------------------|
| 05/22/97 14:35:13 | ENABLE EVENTS CONSOLE ANR2578  |
| 05/30/97 14:29:31 | ENABLE EVENTS CONSOLE ANR0272  |
| 05/30/97 14:31:46 | ENABLE EVENTS USEREXIT ANR0130 |
| 05/30/97 14:31:54 | ENABLE EVENTS USEREXIT ANR0131 |
| 05/30/97 14:50:28 | ENABLE EVENTS USEREXIT ANR0266 |

## Related commands

Table 221. Commands related to QUERY ENABLED

| Command            | Description  |
|--------------------|--|
| BEGIN EVENTLOGGING | Starts event logging to a specified receiver.                |
| DISABLE EVENTS     | Disables specific events for receivers.                      |
| ENABLE EVENTS      | Enables specific events for receivers.                       |
| END EVENTLOGGING   | Ends event logging to a specified receiver.                  |
| QUERY ENABLED      | Displays enabled or disabled events for a specific receiver. |

## QUERY EVENTSERVER (Query the event server)

Use this command to display the name of the event server.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—Query EVENTSERVER—◄◄

### Example: Display the event server name

Display the name of the event server.

```
query eventserver
```

ANR1669I Server EVENT is defined as the event server.

### Related commands

Table 222. Commands related to QUERY EVENTSERVER

| Command            | Description   |
|--------------------|---|
| BEGIN EVENTLOGGING | Starts event logging to a specified receiver.         |
| DEFINE EVENTSERVER | Defines a server as an event server.                  |
| DEFINE SERVER      | Defines a server for server-to-server communications. |
| DELETE EVENTSERVER | Deletes reference to the event server.                |
| DELETE SERVER      | Deletes the definition of a server.                   |
| END EVENTLOGGING   | Ends event logging to a specified receiver.           |

## QUERY EXPORT (Query for active or suspended export operations)

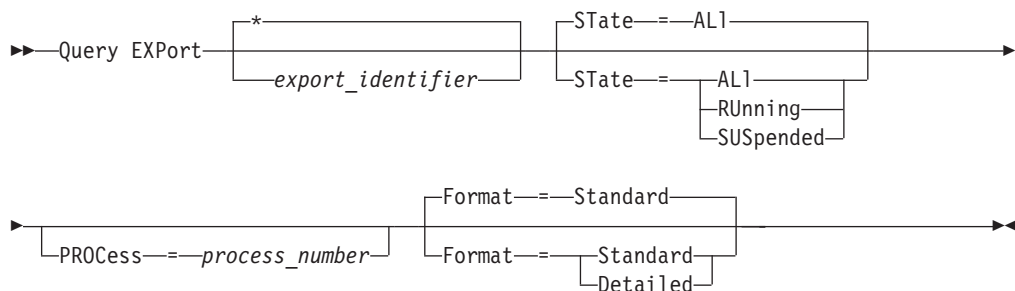
Use this command to list all restartable export operations. A restartable export is a server-to-server export operation whose FILEDATA value is not NONE. Only active server-to-server export operations that can be suspended are displayed.

Any EXPORT NODE or EXPORT SERVER operation with FILEDATA=NONE are not displayed. Additionally, the QUERY EXPORT command does not show export operations where the target device is either sequential media or virtual volumes.

### Privilege class

An administrator can issue this command.

### Syntax



### Parameters

#### *export\_identifier*

This optional parameter is the unique string identifier for the server-to-server export operation. Wildcard characters can be used to specify this name, and all matching export operations are queried. If you do not specify a value for this parameter and you also do not specify a `PROCESS` identifier, then all export operations are queried.

#### **State**

This optional parameter queries the state of the valid server-to-server export operations. The default value is `ALL`. The possible values are:

##### **ALL**

Lists all running and suspended server-to-server export operations.

##### **Running**

Lists all active server-to-server export operations that are identifying eligible files or exporting files to the target server.

##### **SUSPended**

Lists all suspended server-to-server export operations. These suspended operations stopped running because of a failure or by the `SUSPEND EXPORT` command being issued.

#### **PROCEss**

This optional parameter specifies the number of a running server-to-server export operation that you want to query. If `PROCESS` is specified, Tivoli Storage Manager only displays the running server-to-server export operation associated with the process number. If `PROCESS` is not specified, Tivoli Storage Manager displays information on all server-to-server export operations. You



cannot specify this parameter if you specify an export identifier or if you specify the STATE parameter with a value of SUSPENDED.

#### Format

This optional parameter specifies how the information is displayed. The default value is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed for the specified export operations.

#### Detailed

When specified, displays all available information for the export operations.

### Example: Display running and suspended export operations

List information for all currently running and suspended export operations. Issue the following command:

```
query export state=all
```

| Export Identifier | Start Time          | State                   | Process ID | Command   |
|-------------------|---------------------|-------------------------|------------|---|
| MYEXPORTNODE      | 01/24/2007 10:30:03 | Suspended               | --         | Export NODE me,you,them<br>filespace=c\$ nametype=unicode<br>filedata=all durunits=<br>indefinite toserver=athens<br>exportid=MYEXPORTNODE    |
| EXPORT_HOME_DIRS  | 01/25/2007 09:30:03 | Running                 | 11         | Export NODE n2,n3,n4<br>filespace=/home nametype=<br>server filedata=all durunits=<br>indefinite toserver=athens<br>exportid=EXPORT_HOME_DIRS |
| EXPORT_NODE_0001  | 01/25/2007 14:30:33 | Running Not Suspendible | --         | Export NODE n5,n6,n7<br>filespace=d\$ nametype=unicode<br>filedata=archive durunits=<br>indefinite toserver=athens                            |

See “Field descriptions” on page 677 for field descriptions.

### Example: Display information about a running export operation

List information for the currently running export operation with process number “7.” Issue the following command:

```
query export process=7
```

| Export Identifier | Start Time          | State   | Process | Command  |
|-------------------|---------------------|---------|---------|--|
| MYEXPORTNODE      | 01/24/2007 10:30:03 | Running | 7       | Export NODE me,you,them<br>filespace=c\$ nametype=unicode<br>filedata=all toserver=athens<br>exportid=MYEXPORTNODE |

See “Field descriptions” on page 677 for field descriptions.

### Example: Display detailed information about all suspended export operations

List information for all currently suspended export operations. Issue the following command:

```
query export state=suspended format=detailed
```

```

Export Identifier : MyExportNode
Start Time : 01/24/2007 10:30:03
State : Suspended
Process Id : --
Command : Export NODE m* filespace=c$ nametype=unicode
         filedata=all durunits=indefinite
         toserver=athens
Phase : File list complete. Exporting eligible files
Total Running Time : 3 Days 0 Hours 24 Minutes
Current Process Running Time :
Export Operation Restart Count: 0
Date and Time of Last Restart : --
Date and Time of Last Suspend : 01/25/2007 08:30:11
Policy Domains Exported : 0
Policy Sets Exported : 0
Schedules Exported : 0
Mgmt Classes Exported : 0
Copy Groups Exported : 0
Administrators Exported : 1
Option Sets Exported : 0
Node Definitions Exported : 3
Filespace Definitions Exported : 7
Archive Files Exported : 50,000
Backup Files Exported : 150,000
Space Managed Files Exported : 0
Archive Files Skipped : 0
Backup Files Skipped : 25
Space Managed Files Skipped : 0
Total bytes Transferred (MB) : 7,000
Total Files to be Transferred : 900,000
Files Remaining : 700,000

```

See “Field descriptions” on page 677 for field descriptions.

### Example: Display information for server-to-server export operations

List detailed information for all currently running server-to-server export operations. Issue the following command:

```
query export state=running format=detailed
```

```

Export Identifier : export_HOME_Dirs
Start Time : 01/25/2007 09:30:03
State : Running
Process Id : 11
Command : Export NODE n2,n3,n4
         filespace=/home nametype=
         server filedata=all
         toserver=athens
Phase : Identifying and exporting
        eligible files
Total Running Time : 0 Days 22 Hours 0 Minutes
Current Process Running Time : 01:30:00
Export Operation Restart Count: 4
Date and Time of last Restart : 02/01/2007 11:00:03
Date and Time of last Suspend : 01/31/2007 05:01:00
Policy Domains Exported : 0
Policy Sets Exported : 0
Schedules Exported : 0
Mgmt Classes Exported : 0
Copy Groups Exported : 0
Administrators Exported : 1
Option Sets Exported : 0
Node Definitions Exported : 3
Filespace Definitions Exported : 7
Archive Files Exported : 0
Backup Files Exported : 1000
Space Managed Files Exported : 0
Archive Files Skipped : 0
Backup Files Skipped : 0
Space Managed Files Skipped : 0
Total bytes Transferred (MB) : 50
Total Files to be Transferred : 400,000
Files Remaining : 399,000

```

See “Field descriptions” for field descriptions.

## Field descriptions

### Export identifier

The unique identifier assigned to this server-to-server export operation.

### Start time

The time and date that this export operation was first initiated.

**State** The current state of this export operation. The value is one of the following:

#### Running - Not Suspending

The operation is active and is transmitting definitions to the target server. The process cannot be suspended, and if the process fails while in this state, you cannot restart it.

#### Running

The operation is active and is either searching for eligible files or transmitting file data to the target server.

#### Running - Suspend in Progress

The operation is in the process of being suspended as a result of a SUSPEND EXPORT command. The export operation is fully suspended when all of the data from the export operation is saved. An export operation in this state does not respond to the following commands:

- CANCEL PROCESS
- CANCEL EXPORT
- RESTART EXPORT

- SUSPEND EXPORT

### **Suspended**

The operation stopped running due to a failure or was suspended with the SUSPEND EXPORT command.

### **Process ID**

The process ID for the export operation when the status is either "Initializing" or "Running"

### **Command**

The full command issued to start this server-to-server export.

**Phase** The current step that the operation is performing. The possible phases are shown in the order in which they are performed:

#### **Creating definitions on target server**

The operation is exporting definitions. The process cannot be suspended. Should the process fail in this phase, it cannot be restarted.

#### **Identifying and exporting eligible files**

The operation is building a list of eligible files for export. Some files may also be transmitted to the target during this phase. A process in this phase can be suspended. Should the process fail in this phase, it can be restarted.

#### **File list complete. Exporting eligible files**

The operation has completed building the list of eligible files for export and it is now transmitting the files to the target. A process in this phase can be suspended. Should the process fail in this phase, it can be restarted.

### **Total running time**

The overall running time for this server-to-server export operation. For example, if this operation started and was then suspended and restarted two times, this value is the total running time of all three active processes of the export operation.

### **Current process running time**

The running time of the active process of a server-to-server export operation. No value is displayed for a suspended operation because no active process exists.

### **Export operation restart count**

The number of times the server-to-server export operation was restarted.

### **Date and time of last restart**

The last date and time at which this server-to-server export operation was restarted.

### **Date and time of last suspend**

The last date and time at which this server-to-server export operation was suspended.

### **Policy domains exported**

The number of policy domain definitions successfully exported to the target server.

### **Policy sets exported**

The number of policy set definitions successfully exported to the target server.

**Schedules exported**

The number of schedule definitions successfully exported to the target server.

**Mgmt classes exported**

The number of management class definitions successfully exported to the target server.

**Copy groups exported**

The number of copy group definitions successfully exported to the target server.

**Administrators exported**

The number of administrator definitions successfully exported to the target server.

**Option sets exported**

The number of option set definitions successfully exported to the target server.

**Node definitions exported**

The number of node definitions successfully exported to the target server.

**File space definitions exported**

The number of file space definitions successfully exported to the target server.

**Archive files exported**

The number of archive files successfully exported to the target server.

**Backup files exported**

The number of backup files successfully exported to the target server.

**Space managed files exported**

The number of space managed files successfully exported to the target server.

**Archive files skipped**

The number of archive files that were eligible for export but were skipped.

**Backup files skipped**

The number of backup files that were eligible for export but were skipped.

**Space managed files skipped**

The number of space managed files that were eligible for export but were skipped.

**Total bytes transferred (MB)**

The total number bytes transmitted so far to the target server for this export operation.

**Total files to be transferred**

The total number files transmitted so far to the target server for this export operation.

**Files remaining**

The total number files remaining to be transmitted to the target server for this export operation.

### Related commands

*Table 223. Commands related to QUERY EXPORT*

| Command        | Description   |
|----------------|---|
| CANCEL PROCESS | Cancels a background server process.                    |
| CANCEL EXPORT  | Deletes a suspended export operation                    |
| EXPORT NODE    | Copies client node information to external media.       |
| EXPORT SERVER  | Copies all or part of the server to external media.     |
| IMPORT NODE    | Restores client node information from external media.   |
| IMPORT SERVER  | Restores all or part of the server from external media. |
| QUERY PROCESS  | Displays information about background processes.        |
| RESTART EXPORT | Restarts a suspended export operation.                  |
| SUSPEND EXPORT | Suspends a running export operation.                    |

## QUERY FILESPACE (Query one or more file spaces)

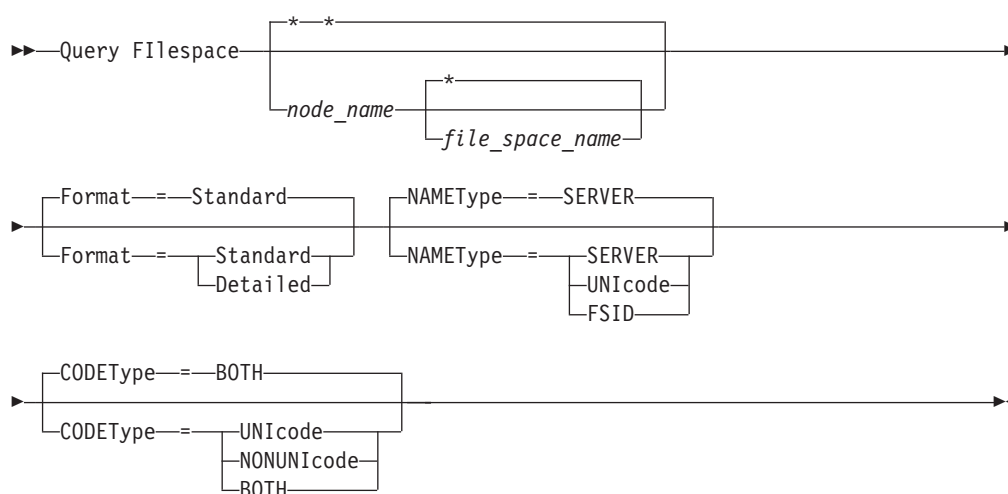
Use this command to display information about file spaces that belong to a client node. The information displayed reflects data as of the last incremental backup.

**Important:** When a node has more than one file space and a DELETE FILESPACE command is issued for only one file space, issuing a QUERY FILESPACE command for the node during the delete process shows no file spaces. When the delete process ends, the remaining file spaces can be viewed with the QUERY FILESPACE command.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies the client node to which the file space belongs. You can use wildcard characters to specify this name. This parameter is optional. The default is all client node names. You must specify a value for this parameter if you specify a file name.

#### *file\_space\_name*

Specifies the name of the file space to be queried. You can use wildcard characters to specify this name. This parameter is optional. If a value is not specified, all file spaces are queried.

For a server that has clients with support for Unicode-enabled file spaces, you may need to have the server convert the file space name that you enter. For example, you may need to have the server convert the name you enter from the server's code page to Unicode. See the **NAMETYPE** parameter for details. If you do not specify a file space name, or specify only a single wildcard character for the name, you can use the **CODETYPE** parameter to limit the operation to Unicode file spaces or to non-Unicode file spaces.

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.

## QUERY FILESPACE

### Format

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.

### NAMEType

Specify how you want the server to interpret the file space names that you enter. This parameter is useful when the server has clients with support for Unicode. You can use this parameter for Unicode-enabled Tivoli Storage Manager clients using Windows, Macintosh OS X, and NetWare operating systems.

Use this parameter only when you enter a partly or fully qualified file space name. The default value is SERVER. Possible values are:

#### SERVER

The server uses the server's code page to interpret the file space names.

#### UNICODE

The server converts the file space name entered from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the name and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

#### FSID

The server interprets the file space names as their file space IDs (FSIDs).

### CODEType

Specify what type of file spaces are to be included in the operation. The default is BOTH, meaning that file spaces are included regardless of code page type. Use this parameter only when you enter a single wildcard character for the file space name. Possible values are:

#### UNICODE

Include only file spaces that are in Unicode.

#### NONUNICODE

Include only file spaces that are not in Unicode.

#### BOTH

Include file spaces regardless of code page type.

## Example: List all file spaces

Query all file spaces associated with all client nodes.

query filesystem

| Node Name | Filespace Name | FSID | Platform | Filespace Type | Is Filespace Unicode? | Capacity (MB) | Pct Util |
|-----------|----------------|------|----------|----------------|-----------------------|---------------|----------|
| JOE       | \\joe\c\$      | 1    | WinNT    | NTFS           | Yes                   | 2,502.3       | 75.2     |
| JOE       | \\joe\d\$      | 2    | WinNT    | NTFS           | Yes                   | 6,173.4       | 59.6     |

See "Field descriptions" on page 683 for field descriptions.



## Example: Display detailed file space information for a virtual file space

Display detailed information for the file space /HomeDir which is a virtual file space mapping and belongs to the NAS node NAS1.

```
query filesystem nas1 /HomeDir
```

| Node Name | Filespace Name | FSID | Platform | Filespace Type | Is Filespace Unicode? | Capacity (MB) | Pct Util |
|-----------|----------------|------|----------|----------------|-----------------------|---------------|----------|
| NAS1      | /HomeDir       | 1    | NetApp   | WAFL (VFS)     | No                    | 2,502.3       | 75.2     |

See “Field descriptions” for field descriptions.

**Important:** The administrator may not see the expected results after requesting a detailed format because several fields have to be filled in by the API application. These include:

- Filespace type
- Platform
- Capacity (MB)
- Pct Util
- Last backup start Date/Time
- Last backup completion Date/Time

For more information on specific fields which are updated by the API, refer to *IBM Tivoli Storage Manager Using the Application Programming Interface*.

## Example: Display detailed file space information for a specific file space and node

Display detailed information on the \\joe\c\$ file space that belongs to the client node JOE.

```
query filesystem joe \\joe\c$ nametype=unicode format=detailed
```

```

Node Name: JOE
Filespace Name: \\joe\c$
Hexadecimal Filespace Name: 5c5c6a6f655c6324
FSID: 1
Platform: WinNT
Filespace Type: NTFS
Is Filespace Unicode?: Yes
Capacity (MB): 2,502.3
Pct Util: 75.2
Last Backup Start Date/Time:
Days Since Last Backup Started:
Last Backup Completion Date/Time:
Days Since Last Backup Completed:

```

See “Field descriptions” for field descriptions.

## Field descriptions

**Important:** The administrator may not see the expected results after requesting a detailed format because several fields must be filled in by the API application. These include:

- Filespace Type

## QUERY FILESPACE

- Platform
- Capacity (MB)
- Pct Util
- Last Backup Start Date/Time
- Last Backup Completion Date/Time

For more information on specific fields that are updated by the API, refer to *IBM Tivoli Storage Manager Using the Application Programming Interface*.

### Node Name

Specifies the name of the client node.

### Filespace Name

Specifies the name of the file space for the client node.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

### Hexadecimal Filespace Name

Specifies the hexadecimal name of the file space for the client node in UTF-8 format.

**FSID** Specifies the file space ID of the file space.

### Platform

Specifies the platform for the client node.

### Filespace Type

Specifies the type of file space.

A file space type that is appended with "(VFS)" denotes that this file space name is a virtual file space mapping for a directory path on a NAS device.

### Is Filespace Unicode?

Indicates whether the file space is Unicode.

### Capacity (MB)

Specifies the amount of space, in megabytes, assigned to this file space on the client node.

For a file space that is a virtual file space mapping for a directory path, this is the capacity of the file space on which the directory path is located.

### Pct Util

Specifies the percentage of the file space that is occupied.

For a file space that is a virtual file space mapping for a directory path, the percentage utilized is calculated as the percentage of the capacity of the file space that was occupied by the directory at the time of the last full backup.

**Last Backup Start Date/Time**

Specifies the start date and time of the last incremental backup of the file space.

**Days Since Last Backup Started**

Specifies the number of days since the start of the last incremental backup of the file space.

**Last Backup Completion Date/Time**

Specifies the completion date and time of the last incremental backup of the file space.

**Days Since Last Backup Completed**

Specifies the number of days since the completion of the last incremental backup of the file space.

**Related commands**

*Table 224. Commands related to QUERY FILESPACE*

| Command                 | Description  |
|-------------------------|--|
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping.   |
| DELETE FILESPACE        | Deletes data associated with client's file spaces.                               |
| REGISTER NODE           | Defines a client to the server and sets options for that user.                   |
| REMOVE NODE             | Removes a client from the list of registered nodes for a specific policy domain. |
| RENAME FILESPACE        | Renames a client filesystem on the server.                                       |
| UPDATE NODE             | Changes the attributes associated with a client node.                            |

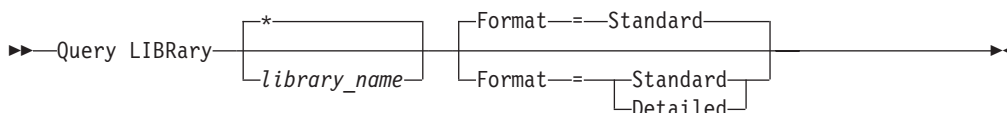
## QUERY LIBRARY (Query a library)

Use this command to display information about libraries.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *library\_name*

Specifies the name of the library to be queried. You can use wildcards to specify names. This parameter is optional.

#### Format

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.

### Example: Display summary information about a specific library

Display information about the library named AUTO. Issue the command:

```
query library auto
```

```

Library Name: AUTO
Library Type: SCSI
ACS Id:
Private Category:
Scratch Category:
WORM Scratch Category:
External Manager:
Shared: No
LanFree:
ObeyMountRetention:
    
```

See “Field descriptions” on page 687 for field descriptions.

### Example: Display detailed library information about a specific library

Display information in full detail about the library named EZLIFE. Issue the command:

```
query library ezlife format=detailed
```

```

Library Name: EZLIFE
Library Type: SCSI
ACS Id:
Private Category:
Scratch Category:
WORM Scratch Category:
External Manager:
Shared: Yes
LanFree:
ObeyMountRetention:
Primary Library Manager: EZSERVER
WWN:
Serial Number:
AutoLabel: OVERWRITE
Relabel Scratch: Yes
Last Update by (administrator): DOCTOR_MIKE
Last Update Date/Time: 2002-12-05 15:24:53

```

See “Field descriptions” for field descriptions.

## Field descriptions

### Library Name

The name of the library.

### Library Type

The type of library.

### ACS Id

Specifies that the library is a StorageTek library that is controlled by StorageTek Automated Cartridge System Library Software (ACSL).

### Private Category

The category number for private volumes that must be mounted by name.

The information displayed in this field only applies to an IBM 3494 or 3495 Tape Library Dataserver.

### Scratch Category

The category number to use for scratch volumes in the library.

The information displayed in this field only applies to an IBM 3494 or 3495 Tape Library Dataserver.

### WORM Scratch Category

The category number used for WORM scratch volumes in the library.

The information displayed in this field only applies to an IBM 3494 or 3495 Tape Library Dataserver.

### External Manager

The location of the external library manager where the server can send media access requests.

### Shared

Whether this library is shared with other Tivoli Storage Manager servers in a storage area network (SAN).

### LanFree

Whether an external library is used for LAN-free operations.

### ObeyMountRetention

Whether the server uses the value set for mount retention in the device class associated with this external library.

## QUERY LIBRARY

### Primary Library Manager

The name of the server that is responsible for controlling access to library resources.

**WWN** The Fibre Channel worldwide name for the library.

### Serial Number

Specifies the serial number for the library being queried.

### AutoLabel

Specifies whether the server attempts to automatically label tape volumes.

### Reset Drives

Specifies whether the server performs a target reset when the server is restarted or when a library client or storage agent re-connection is established.

### Relabel Scratch

Specifies whether the server relabels volumes that have been deleted and returned to scratch.

### Last Update by (administrator)

Who performed the last update to the library.

### Last Update Date/Time

The date and time when the last update occurred.

## Related commands

*Table 225. Commands related to QUERY LIBRARY*

| Command        | Description   |
|----------------|---|
| AUDIT LIBRARY  | Ensures that an automated library is in a consistent state.         |
| DEFINE LIBRARY | Defines an automated or manual library.                             |
| DEFINE PATH    | Defines a path from a source to a destination.                      |
| DELETE LIBRARY | Deletes a library.  |
| QUERY PATH     | Displays information about the path from a source to a destination. |
| UPDATE LIBRARY | Changes the attributes of a library.                                |
| UPDATE PATH    | Changes the attributes associated with a path.                      |

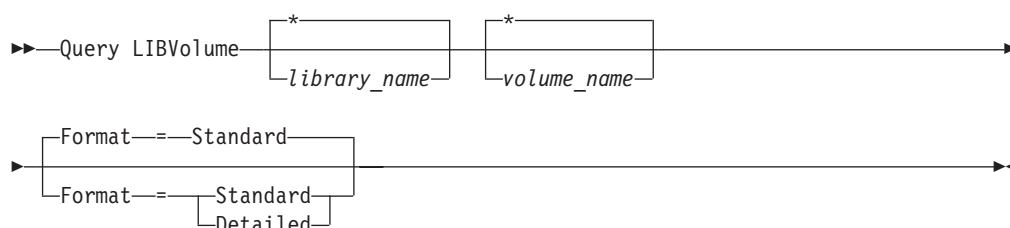
## QUERY LIBVOLUME (Query a library volume)

Use this command to display information about one or more volumes that are checked into an automated library for use by the Tivoli Storage Manager server.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *library\_name*

Specifies the name of the library. You can use wildcard characters to specify this name. This parameter is optional. The default is all libraries.

#### *volume\_name*

Specifies the volume name. You can use wildcard characters to specify this name. This parameter is optional. The default is all volumes.

#### Format

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.

##### **Detailed**

Specifies that complete information is displayed.

### Example: List checked in volumes for a specific library

Display information about all of the volumes that are checked into the library named TAPE. See “Field descriptions” on page 690 for field descriptions.

```
query libvolume tape
```

| Library Name | Volume Name | Status  | Owner | Last Use | Home Element | Device Type |
|--------------|-------------|---------|-------|----------|--------------|-------------|
| TAPE         | 000114      | Scratch |       |          | 1,000        | LTO         |
| TAPE         | NY1602      | Scratch |       |          | 1,001        | DLT         |

### Example: Display detailed information for a specific library

Display detailed information about a volume named JJY008. See “Field descriptions” on page 690 for field descriptions.

```
query libvolume jjy008 format=detailed
```

## QUERY LIBVOLUME

```
Library Name: HPW3494
Volume Name: JJY008
Status: Private
Owner: SUNSET
Last Use: Data
Home Element:
Device Type:
Cleanings Left:
Media Type:
```

### Field descriptions

#### Library Name

The name of the library where the storage volume is located.

#### Volume Name

The name of the storage volume.

**Status** The status of the storage volume according to the library inventory. If the status is Private, the volume is being used by Tivoli Storage Manager. If the status is Scratch, the volume is available for use.

#### Owner

The owner server of the volume, if the volume is private.

#### Last Use

The type of data on the volume. This field applies only to volumes in Private status. For storage pool volumes, this field shows **Data**. For database backup volumes (full, incremental, or snapshot), this field shows **DbBackup**.

#### Home Element

The element address of the library slot containing the volume.

#### Device Type

The type of device that the volume is used on. This field will display a value only for volumes checked into a library that has mixed media capabilities.

#### Cleanings Left

For cleaner cartridges, the number of cleanings left.

#### Media Type

The type of media the volume represents (for example, 8mm tape).

### Related commands

Table 226. Commands related to QUERY LIBVOLUME

| Command            | Description  |
|--------------------|--|
| AUDIT LIBRARY      | Ensures that an automated library is in a consistent state.              |
| CHECKIN LIBVOLUME  | Checks a storage volume into an automated library.                       |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library.                     |
| DEFINE VOLUME      | Assigns a volume to be used for storage within a specified storage pool. |



Table 226. Commands related to QUERY LIBVOLUME (continued)

| Command          | Description                                       |
|------------------|---|
| LABEL LIBVOLUME  | Labels volumes in manual or automated libraries.  |
| QUERY LIBRARY    | Displays information about one or more libraries. |
| UPDATE LIBVOLUME | Changes the status of a storage volume.           |

### QUERY LICENSE (Display license information)

Use this command to display license audit, license terms, and compliance information.

#### Privilege class

Any administrator can issue this command.

#### Syntax

►►—Query LICense—◄◄

#### Parameters

None.

#### Related commands

*Table 227. Commands related to QUERY LICENSE*

| Command                | Description   |
|------------------------|---|
| AUDIT LICENSES         | Checks for compliance with defined licenses.  |
| QUERY AUDITOCCUPANCY   | Displays the server storage utilization for a client node.                              |
| QUERY STATUS           | Displays the settings of server parameters, such as those selected by the SET commands. |
| REGISTER LICENSE       | Registers a new license with the IBM Tivoli Storage Manager server.                     |
| SET LICENSEAUDITPERIOD | Specifies the number of days between automatic license audits.                          |

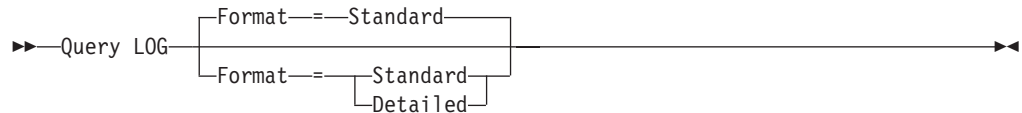
# QUERY LOG (Display information on the recovery log)

Use this command to display information about the recovery log.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

### Format

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed.

#### Detailed

Specifies that complete information is displayed.

## Example: Display summary information about the recovery log

Display summary information about the recovery log. See “Field descriptions” for field descriptions.

query log

| Total Space<br>(MB) | Used Space<br>(MB) | Free Space<br>(MB) |
|---------------------|--------------------|--------------------|
| -----               | -----              | -----              |
| 38,912              | 401.3              | 38,368.7           |

## Example: Display detailed information about the recovery log

Display detailed information about the recovery log. See “Field descriptions” for field descriptions.

query log format=detailed

```

Total Space (MB): 38,912
Used Space (MB): 401.3
Free Space (MB): 38,368.7
Active Log Directory : /activelog
Mirror Log Directory : /mirrorlog
Archive Failover Log Directory : /archfailoverlog
Archive Log Directory : /archivelog
    
```

## Field descriptions

### Total Space

Specifies the maximum size of the active log, in megabytes.

### Used Space

Specifies the amount of used active log space, in megabytes.

## QUERY LOG

### **Free Space**

Specifies the amount of active log space that is not being used by uncommitted transactions, in megabytes.

### **Active Log Directory**

Specifies the location where active log files are stored. When you change the active log directory, the server moves all archived logs to the archive log directory and all active logs to a new active log directory.

### **Mirror Log Directory**

Specifies the location where the mirror for the active log is maintained.

### **Archive Failover Log Directory**

Specifies the location into which the server saves archive logs if the logs cannot be archived to the archive log directory.

### **Archive Log Directory**

Specifies the location into which the server can archive a log file after all the transactions that are represented in that log file are completed.

## QUERY MACHINE (Query machine information)

Use this command to display information for one or more machines. You can use this information to recover Tivoli Storage Manager client machines in case of a disaster.

**Attention:** Tivoli Storage Manager does not use the information in any way. It is available only to help you plan for the disaster recovery of client machines.

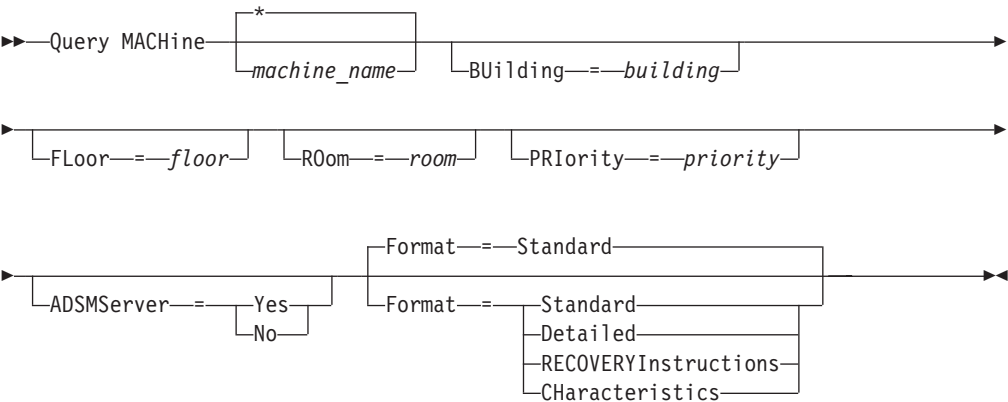
Tivoli Storage Manager displays information for multiple machines in the following order:

- According to the priority specified.
- Within a priority, according to the specified location and machine name.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

*machine\_name*  
Specifies the name of one or more machines to be queried. You can use wildcard characters to specify this name. This parameter is optional. The default is all machines that meet the specified criteria.

**BUilding**  
Specifies the name or number of the building that the machines are in. This parameter is optional. Enclose the text in quotation marks if it contains any blank characters.

**FLoor**  
Specifies the name or number of the floor that the machines are on. This parameter is optional. Enclose the text in quotation marks if it contains any blank characters.

**ROom**  
Specifies the name or number of the room that the machines are in. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters.

### PRIority

Specifies the priority number of the machines. This parameter is optional.

### ADSMServer

Specifies if the machine contains a Tivoli Storage Manager server. This parameter is optional. The default is to display any machines that meet the other criteria. Possible values are:

#### Yes

The machine contains a Tivoli Storage Manager server.

#### No

The machines do not contain a Tivoli Storage Manager server.

### Format

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. Possible values are:

#### Standard

Tivoli Storage Manager displays partial information for the machines.

#### Detailed

Tivoli Storage Manager displays all information for the machines.

### RECOVERYInstructions

Tivoli Storage Manager displays only machine recovery instructions. This option is valid only when querying a specific machine.

### CHaracteristics

Tivoli Storage Manager displays only machine characteristics. This option is valid only when querying a specific machine.

## Example: Display information for a specific machine

Display information for a machine named MACH1. See “Field descriptions” on page 697 for field descriptions.

```
query machine MACH1
```

| Machine<br>Name | Machine<br>Priority | Building | Floor | Room | Node Name | Recovery<br>Media Name |
|-----------------|---------------------|----------|-------|------|-----------|------------------------|
| MACH1           | 1                   | 21       | 2     | 2929 | VIRGINIA  | RECMED1                |

## Example: Display detailed information for priority 1 machines

Display detailed information for all priority 1 machines on the second floor of building 21. See “Field descriptions” on page 697 for field descriptions.

```
query machine * building=21 floor=2 priority=1
format=detailed
```

```
Machine Name: MACH1
Machine Priority: 1
Building: 21
Floor: 2
Room: 2929
Server?: Yes
Description: TSM server machine
Node Name: VIRGINIA
Recovery Media Name: RECMED1
Characteristics?: Yes
Recovery Instructions?: Yes
```

## Field descriptions

### Machine Name

The name of the machine.

### Machine Priority

The recovery priority of the machine.

### Building

The building in which the machine is located.

**Floor** The floor on which the machine is located.

**Room** The room in which the machine is located.

### Server?

Whether the machine contains a Tivoli Storage Manager server.

### Description

A description of the machine.

### Node Name

The Tivoli Storage Manager client nodes associated with this machine.

### Recovery Media Name

The recovery media associated with this machine.

### Characteristics?

Whether the characteristics text of the machine is stored in the database.

### Recovery Instructions?

Specifies whether recovery instructions text for a machine is stored in the Tivoli Storage Manager database.

## Related commands

Table 228. Commands related to QUERY MACHINE

| Command                      | Description  |
|------------------------------|--|
| DEFINE MACHINE               | Defines a machine for DRM.   |
| DEFINE MACHNODEASSOCIATION   | Associates an IBM Tivoli Storage Manager node with a machine.  |
| DEFINE RECMEDMACHASSOCIATION | Associates recovery media with a machine.  |
| DELETE MACHINE               | Deletes a machine.   |
| INSERT MACHINE               | Inserts machine characteristics or recovery instructions into the IBM Tivoli Storage Manager database. |
| UPDATE MACHINE               | Changes the information for a machine.   |

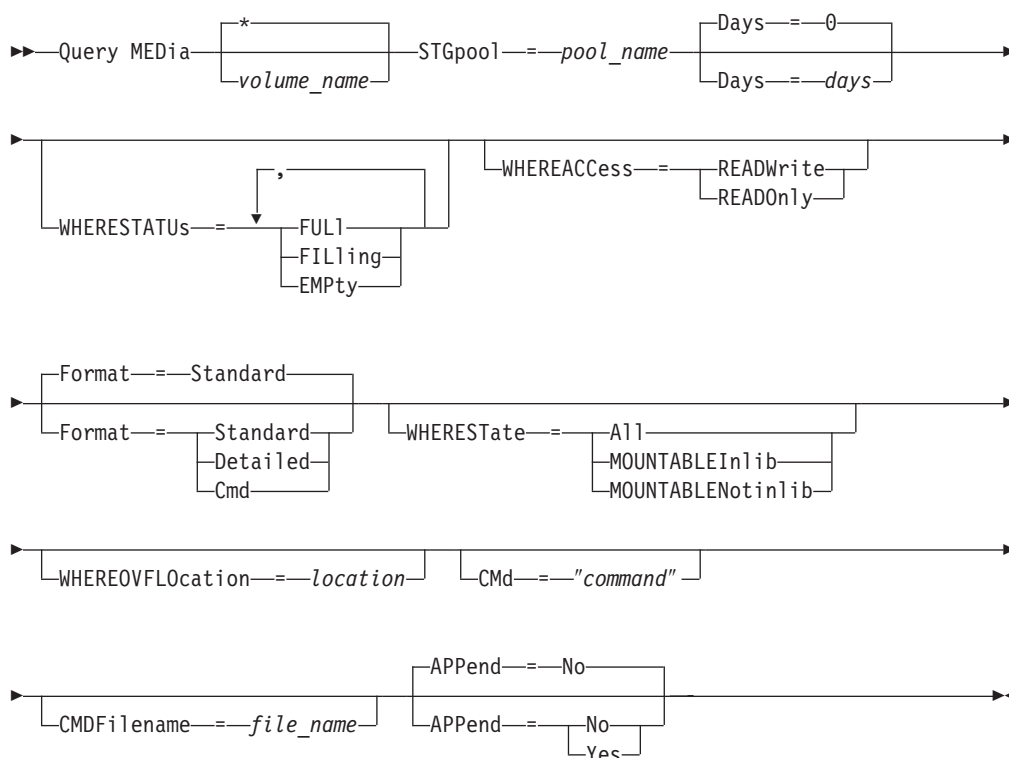
## QUERY MEDIA (Query sequential access storage pool media)

Use this command to display information about the sequential access primary and copy storage pool volumes moved by the MOVE MEDIA command.

### Privilege class

Any administrator with system or operator privilege can issue this command unless it includes the CMD parameter. If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO, the administrator must have operator, unrestricted storage, or system privilege. If the CMD parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES (the default), the administrator must have system privilege.

### Syntax



### Parameters

#### *volume\_name*

Specifies the name of the sequential access primary or copy storage pool volume to display. This parameter is optional. You can use a wildcard character to specify the name. All matching volumes are considered for processing. If you do not specify this parameter, all volumes defined in the storage pool specified with the STGPOOL parameter display.

#### **STGpool (Required)**

Specifies the name of the sequential access primary or copy storage pool that is used to select the volumes for processing. You can use wildcard characters to specify the name. All matching storage pools are processed. If the storage pool specified is not managed by an automated library, no volumes display.



### **Days**

Specifies the number of days that must elapse, after the volume has been written to or read from, before the volume is eligible for processing. This parameter is optional. You can specify a number from 0 to 9999. The default value is 0. The most recent of the volume's last written date or last read date is used to calculate the number of days elapsed.

### **WHERESTATUS**

Specifies that the output of the query should be restricted by volume status. This parameter is optional. You can specify more than one status in a list by separating each status with a comma and no intervening spaces. If you do not specify a value for this parameter, all volumes in the specified storage pool, regardless of their status, are displayed.

Possible values are:

#### **FULL**

Specifies that volumes with a status of FULL display.

#### **FILLing**

Specifies that volumes with a status of FILLING display.

#### **EMPTy**

Specifies that volumes with a status of EMPTY display.

### **WHEREACcEss**

Specifies that output should be restricted by volume access mode. This parameter is optional. 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 display.

#### **READOnly**

Specifies that volumes with an access mode of READONLY display.

### **Format**

Specifies how information displays. This parameter is optional. The default value is STANDARD. Possible values are:

#### **Standard**

Specifies that partial information displays for the specified sequential access storage pool volumes.

#### **Detailed**

Specifies that complete information displays for the specified sequential access storage pool volumes.

#### **Cmd**

Specifies that executable commands are built for the storage pool volumes processed by the QUERY MEDIA command. These commands will be in the file specified with the CMDFILENAME parameter on the QUERY MEDIA command. If you want the commands to display on the console only, specify a null string (") for the CMDFILENAME. If FORMAT=CMD is specified but no command string is specified with the CMD parameter, the QUERY MEDIA command will fail.

### **WHEREState**

Specifies the state of volumes to process. This parameter restricts processing to volumes that have the specified state. This parameter is optional. The default is ALL. Possible values are:

### All

Specifies that volumes in all states are queried. The valid states are: MOUNTABLEINLIB and MOUNTABLENOTINLIB.

### MOUNTABLEInlib

Specifies that volumes that are currently in the MOUNTABLEINLIB state are queried. Volumes in the MOUNTABLEINLIB state are in the library, and are onsite, contain valid data, and are available for onsite processing.

### MOUNTABLENotinlib

Specifies that volumes that are currently in the MOUNTABLENOTINLIB state are queried. Volumes in the MOUNTABLENOTINLIB state are not in the library, do not contain valid data, and are not available for onsite processing.

### WHEREOVFLocation

Specifies the overflow location of the volumes to display. This parameter is optional. This parameter restricts processing to volumes that are in the specified location. The maximum length of the location is 255 characters. The location must be enclosed in quotation marks if it contains any blank characters.

### CMd

Specifies the creation of executable commands. Enclose the command specification in quotation marks. The maximum length of the command specification is 255 characters. This parameter is optional.

For each volume successfully processed by the QUERY MEDIA command, the server writes the associated commands to a file. Specify the file name with the CMDFILENAME parameter.

If you do not specify a filename, the command will generate a default filename by appending the string "exec.cmds.media" to the server directory.

### Remember:

1. If the command written to the file exceeds 255 characters, it is split into multiple lines, and a continuation character (+) is added to all but the last line. You may need to alter the continuation character according to the requirements of the product that runs the commands.
2. If an executable command is specified with any value for FORMAT other than CMD, the command string is ignored, and the QUERY MEDIA command will not write any command line.

Specify a command string and any substitution variables:

#### *string*

Specifies the string to build an executable command to process the volume name or volume location or both. You can specify any free form text for the string. Do not use embedded quotation marks. For example, the following is a valid executable command specification:

```
cmd="checkin libvolume &vol"
```

The following is an invalid executable command specification:

```
cmd="checkin libvolume "&vol""
```

#### *substitution*

Specifies a variable for which you want the QUERY MEDIA command to substitute a value. The possible substitution variables are:

### **&VOL**

Substitute the volume name for &VOL. You can specify lowercase characters, &vol. No spaces or blanks are allowed between ampersand, &, and VOL. If there are spaces or blanks between ampersand and VOL, the QUERY MEDIA command will treat them as strings and no substitution will be set. If &VOL is not specified, no volume name is set in the executable command.

### **&LOC**

Substitute the volume location for &LOC. You can specify lowercase characters, &loc. No spaces or blanks are allowed between ampersand, &, and LOC. If there are spaces or blanks between ampersand and LOC, the QUERY MEDIA command will treat them as strings and no substitution will be set. If &LOC is not specified, no location name is set in the executable command.

### **&VOLDSN**

Substitute the volume file name for &VOLDSN. An example of a copy storage pool tape volume file name using the defined prefix IBM Tivoli Storage Manager310 is IBM Tivoli Storage Manager310.BFS. If &VOLDSN is not specified, no volume file name is set in the executable command.

### **&NL**

Substitute the new line character for &NL. When &NL is specified, the QUERY MEDIA command will split the command at the position where the &NL is and will not append any continuation character. The user is responsible for specifying the proper continuation character before the &NL if one is required. The user is also responsible for the length of the line written. If the &NL is not specified and the command exceeds 255 characters, the command is split into multiple lines, and a continuation character (+) is added to all but the last line.

### **CMDFilename**

Specifies the full path name that will contain the commands specified with CMD parameter when FORMAT=CMD is specified. This parameter is optional. The maximum length of the file name is 1279 characters.

If you specify "" with the CMDFILENAME parameter, the QUERY MEDIA command will generate a file name by appending the "exec.cmds.media" to the server directory. The server directory is the current working directory of the server process.

If you specify a null string ("") for the CMDFILENAME, the commands built are displayed on the console only. You can redirect the commands displayed to a file by using the redirection characters for the operating system (> or >>).

If the filename is not specified, the command will generate a default filename by appending the string "exec.cmds.media" to the server directory.

The QUERY MEDIA command automatically allocates the file name specified or generated. If the file name exists, the QUERY MEDIA command will attempt to use it and the existing data, if any, in the file to be overwritten. You can specify APPEND=YES to prevent the existing data from being overwritten. If the QUERY MEDIA command fails after the command file is allocated, the file is not deleted.

### **APPend**

Specifies to write at the beginning or the ending of the command file data. This parameter is optional. The default is NO. Possible values are:

### No

Specifies to write the data from the beginning of the command file. If the given command file exists, its contents are overwritten.

### Yes

Specifies to append the command file by writing at the end of the command file data.

### Example: Display information on a specific sequential access storage pool

Display all full and partial full volumes that are in the sequential access primary storage pool, ARCHIVE. See “Field descriptions” on page 703 for field descriptions.

```
query media * stgpool=archive wherestatus=full, filling
```

| Volume Name | State                 | Location        | Automated LibName |
|-------------|-----------------------|-----------------|-------------------|
| TAPE01      | Mountable in Library  |                 |                   |
| TAPE03      | Mountable not in Lib. | Room1234/Bldg31 | LIB3494           |
| TAPE07      | Mountable in Library  |                 | LIB3494           |
| TAPE09      | Mountable not in Lib. | Room1234/Bldg31 |                   |

### Example: Display information on sequential access storage pool with a specific prefix

Display in detail all full volumes in MOUNTABLENOTINLIB state for sequential access storage pools that have a prefix name of ONSITE. See “Field descriptions” on page 703 for field descriptions.

```
query media wherestate=mountablenotinlib stgpool=onsite*
wherestatus=full format=detailed
```

```

Volume Name: TAPE21
State: Mountable not in library
Status: Full
Access: ReadOnly
Last Reference Date: 01/30/98
Last Update Date/Time: 08/20/1996 13:29:02
Location: Rm569/bldg31
Storage Pool Name: ONSITE.ARCHIVE
Automated Libname:

Volume Name: TAPE22
State: Mountable not in library
Status: Full
Access: ReadOnly
Last Reference Date: 01/30/98
Last Update Date/Time: 08/20/1996 15:29:02
Location: Rm569/bldg31
Storage Pool Name: ONSITE.ARCHIVEPOOL
Automated Libname:
```

### Example: Generate checkin commands

Generate the CHECKIN LIBVOLUME commands for full and partially full volumes that are in the ONSITE.ARCHIVE primary storage pool and stored in the overflow location Room 2948/Bldg31.

```
query media * stgpool=onsite.archive format=cmd
wherestatus=full,filling wherestate=mountablenotinlib
whereovflocation=room2948/bldg31
cmd="checkin libvol lib3494 &vol status=private"
cmdfilename=/tsm/move/media/checkin.vols
```

The QUERY MEDIA command created the CHECKIN LIBVOLUME executable commands in /tsm/move/media/checkin.vols, which can be run by issuing the MACRO command with /tsm/move/media/checkin.vols as the macro name.

```
checkin libvol lib3494 TAPE04 status=private
checkin libvol lib3494 TAPE13 status=private
checkin libvol lib3494 TAPE14 status=private
```

## Field descriptions

### Volume Name

Specifies the name of the primary sequential access storage pool volume.

**State** Specifies the state of the volume.

**Status** Specifies the status of the volume.

### Access

Specifies the access mode of the volume.

### Last Reference Date

Specifies the volume's last written date or last read date, whichever is more recent.

### Last Update Date/Time

Specifies the date and time when the volume was most recently updated.

### Location

Specifies where the volume is stored. If the volume is ejected from the library and its location is not specified or defined, a question mark (?) is displayed for the location.

### Storage Pool Name

Specifies the name of the sequential access storage pool where the volume is defined.

### Automated LibName

Specifies the automated library name if the volume is in the library.

## Related commands

Table 229. Commands related to QUERY MEDIA

| Command    | Description  |
|------------|--|
| MOVE MEDIA | Moves storage pool volumes that are managed by an automated library. |

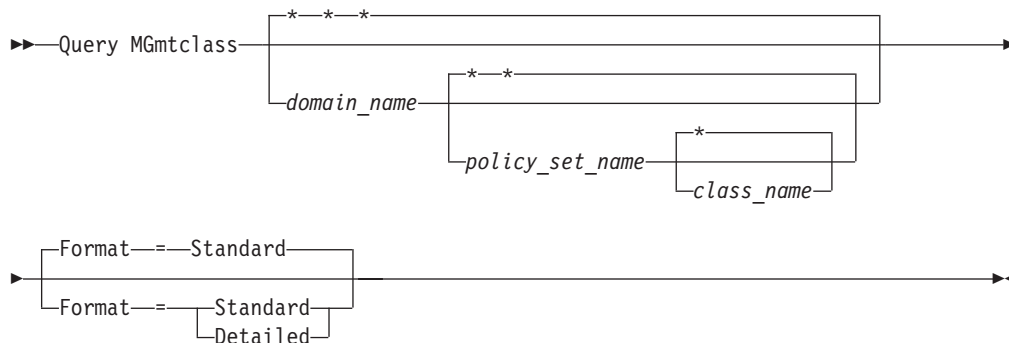
## QUERY MGMTCLASS (Query a management class)

Use this command to display information about management classes.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *domain\_name*

Specifies the policy domain associated with the management class to query. This parameter is optional. You can use wildcard characters to specify this name. If you do not specify a value for this parameter, management classes in all policy domains are queried. You must specify this parameter when querying an explicitly named management class.

#### *policy\_set\_name*

Specifies the policy set associated with the management class to query. This parameter is optional. You can use wildcard characters to specify this name. If you do not specify a value for this parameter, management classes in all policy sets are queried. You must specify this parameter when querying an explicitly named management class.

#### *class\_name*

Specifies the management class to query. This parameter is optional. You can use wildcard characters to specify this name. If you do not specify a value for this parameter, all management classes are queried.

#### Format

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.

##### **Detailed**

Specifies that complete information is displayed.

### Example: Display information for all management classes

Query all management classes for all policy domains. Create the output in standard format. See "Field descriptions" on page 705 for field descriptions.

```
query mgmtclass
```

| Policy Domain Name | Policy Set Name | Mgmt Class Name | Default Mgmt Class ? | Description                        |
|--------------------|-----------------|-----------------|----------------------|------------------------------------|
| EMPLOYEE-RECORDS   | ACTIVE          | ACTIVEFILES     | Yes                  | Modified default management class  |
| EMPLOYEE-RECORDS   | HOLIDAY         | ACTIVEFILES     | Yes                  | Modified default management class  |
| EMPLOYEE-RECORDS   | HOLIDAY         | FILEHISTORY     | No                   | Test modified management class     |
| EMPLOYEE-RECORDS   | VACATION        | ACTIVEFILES     | Yes                  | Original default management class  |
| EMPLOYEE-RECORDS   | VACATION        | FILEHISTORY     | No                   | Test modified management class     |
| 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 |

### Example: Display detailed information for a specific management class

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. See “Field descriptions” for field descriptions.

```
query mgmtclass employee_records vacation
activefiles format=detailed
```

```

Policy Domain Name: EMPLOYEE_RECORDS
Policy Set Name: VACATION
Mgmt Class Name: ACTIVEFILES
Default Mgmt Class ? : Yes
Description: Installed default management class
Space Management Technique: None
Auto-Migrate on Non-Use: 0
Migration Requires Backup?: Yes
Migration Destination: SPACEMGPPOOL
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 05/31/1998 13:15:45
Managing Profile: EMPLOYEE
Changes Pending: Yes
```

### Field descriptions

#### Policy Domain Name

The policy domain.

#### Policy Set Name

The policy set.

#### Mgmt Class Name

The management class.

#### Default Mgmt Class?

Whether the management class is the default management class for the policy set.

#### Description

The description of the management class.

## QUERY MGMTCLASS

### Space Management Technique

The space management technique for the management class, for Tivoli Storage Manager for Space Management clients.

### Auto-Migrate on Non-Use

The number of days that must elapse since a file was last accessed before it is eligible for automatic migration by Tivoli Storage Manager for Space Management clients.

### Migration Requires Backup?

Whether a backup version of a file must exist before a file can be migrated by Tivoli Storage Manager for Space Management clients.

### Migration Destination

The storage pool that is the destination for files migrated by Tivoli Storage Manager for Space Management clients.

### Last Update by (administrator)

The administrator or server that most recently updated the management class. If this field contains \$\$CONFIG\_MANAGER\$\$, the management class is associated with a domain that is managed by the configuration manager.

### Last Update Date/Time

The date and time when the management class was most recently defined or updated.

### Managing Profile

The profile or profiles to which the managed server subscribed to get the definition of this management class.

### Changes Pending

Whether or not changes are being made but not activated. Once the changes are activated, the field resets to No.

## Related commands

Table 230. Commands related to QUERY MGMTCLASS

| Command                | Description   |
|------------------------|---|
| COPY MGMTCLASS         | Creates a copy of a management class.   |
| DEFINE MGMTCLASS       | Defines a management class.   |
| DEFINE PROFASSOCIATION | Associates objects with a profile.  |
| DELETE MGMTCLASS       | Deletes a management class and its copy groups from a policy domain and policy set. |
| QUERY DOMAIN           | Displays information about policy domains.  |
| UPDATE MGMTCLASS       | Changes the attributes of a management class.                                       |



## QUERY MOUNT (Display information on mounted sequential access volumes)

Use this command to display information about the status of one or more sequential access volumes that are mounted.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

*volume\_name*

Specifies the name of the mounted sequential access volume. You can use wildcard characters to specify this name. This parameter is optional. The default is *all mounted volumes*.

### Example: List all mounted sequential volumes

Display information on all mounted sequential media volumes.

query mount

```

ANR8330I 3590 volume D6W992 is mounted R/O
in drive RMT1(/dev/rmt1), status: IN USE.
ANR8334I 1 volumes found.
ANR8331I 8MMTAPE volume WPD000 is mounted R/W
in drive 8MM.1 (/dev/mt0), status: DISMOUNTING.
ANR8334I 1 volumes found.
  
```

### Remember:

1. If the status of a volume is full or if its access mode is read-only (R/O), the mount mode of the volume will be R/O. To determine the status and access mode of a volume, issue the `QUERY VOLUME FORMAT=DETAILED` command. If a volume can be written to (that is, the status is filling or empty), the mount mode of the volume will be read/write (R/W), even if it is currently only being read.
2. In a storage pool associated with the FILE device type, the server can perform concurrent multiple read-access and one write-access to the same volume. As a result, a volume in a storage pool with a device type of FILE can appear as mounted more than once.

### Related commands

Table 231. Commands related to QUERY MOUNT

| Command         | Description  |
|-----------------|--|
| DISMOUNT VOLUME | Dismounts a sequential, removable volume by the volume name. |
| REPLY           | Allows a request to continue processing.                     |

## QUERY NASBACKUP (Query NAS backup images)

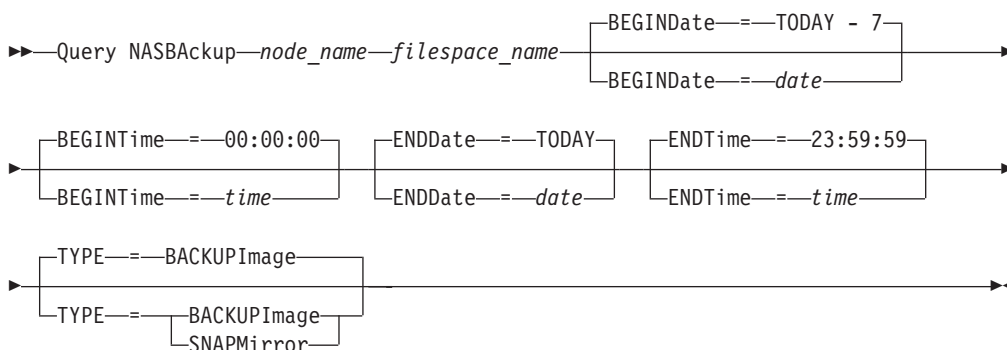
Use this command to display information about the file system image objects that have been backed up for a specific NAS node and file space. You can only use this command to display objects that were backed up for a NAS node using NDMP.

The server displays all matching objects, the dates that these objects were backed up, and information about a table of contents (TOC) for the object.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### node\_name (Required)

Specifies the name of the NAS node for which backup objects are displayed. You cannot use wildcards to specify this name.

#### filespace\_name (Required)

Specifies the name of the file space for which backup objects are displayed. You can use wildcards to specify this name.

#### BEGINDate

Specifies the beginning date to select the backup objects to display. All backup objects that were created on or after the specified date are displayed. The default is seven days prior to the current date. You can use this parameter with the BEGINTIME parameter to specify a range for the date and time. This parameter is optional.

You can specify the date using one of the following values:

| Value               | Description  | Example  |
|---------------------|--|--|
| MM/DD/YYYY          | A specific date  | 09/15/2002   |
| TODAY               | The current date   | TODAY  |
| TODAY-days or -days | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY -7 or -7.<br><br>To display information about the image objects that have been created a week ago, you can specify<br>BEGINDATE=TODAY-7 or<br>BEGINDATE= -7. |

### BEGINTime

Specifies the beginning time to select the backup objects to display. All backup objects created on or after the specified time display. This parameter is optional. The default is midnight (00:00:00) on the date specified for the BEGINDATE.

You can specify the time using one of the following values:

| Value                                  | Description  | Example  |
|--|--|--|
| <i>HH:MM:SS</i>                        | A specific time on the specified begin date                          | 10:30:08   |
| NOW                                    | The current time on the specified begin date                         | NOW  |
| NOW+ <i>HH:MM</i> or<br>+ <i>HH:MM</i> | The current time plus hours and minutes on the specified begin date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 with BEGINTIME=NOW+3 or BEGINTIME=+3, Tivoli Storage Manager displays image objects with a time of 12:00 or later on the begin date.       |
| NOW- <i>HH:MM</i> or<br>- <i>HH:MM</i> | The current time minus hours and minutes on the specified begin date | NOW-04:00 or -04:00.<br><br>If you issue this command at 9:00 with BEGINTime=NOW-3:30 or BEGINTime= -3:30, Tivoli Storage Manager displays image objects with a time of 5:30 or later on the begin date. |

### ENDDate

Specifies the ending date used to select the backup objects to be displayed. All backup objects created on or before the specified date are displayed. This parameter is optional. The default is the current date. You can use this parameter with the ENDTIME parameter to specify an ending date and time.

You can specify the date using one of the following values:

| Value                                  | Description  | Example  |
|--|--|--|
| <i>MM/DD/YYYY</i>                      | A specific date  | 09/15/2002   |
| TODAY                                  | The current date   | TODAY  |
| TODAY- <i>days</i> or<br>- <i>days</i> | The current date minus days specified. The maximum number of days you can specify is 9999. | TODAY-1 or -1.<br><br>To display information created up to yesterday, you can specify ENDDATE=TODAY-1 or simply ENDDATE= -1. |

### ENDTime

Specifies the ending time used to select the backup objects to be displayed. All backup objects created on or before the specified time are displayed. This parameter is optional. The default is 23:59:59. You can use this parameter with the ENDDATE parameter to specify a range for the date and time.

## QUERY NASBACKUP

You can specify the time using one of the following values:

| Value                      | Description  | Example   |
|----------------------------|--|---|
| <i>HH:MM:SS</i>            | A specific time on the specified end date                          | 10:30:08  |
| NOW                        | The current time on the specified end date                         | NOW   |
| <i>NOW+HH:MM or +HH:MM</i> | The current time plus hours and minutes on the specified end date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 with <code>ENDTIME=NOW+3:00</code> or <code>ENDTIME= +3:00</code> , Tivoli Storage Manager displays image objects with a time of 12:00 or later on the end date you specify.    |
| <i>NOW-HH:MM or -HH:MM</i> | The current time minus hours and minutes on the specified end date | NOW-03:30 or -03:30.<br><br>If you issue this command at 9:00 with <code>ENDTIME=NOW-3:30</code> or <code>ENDTIME= -3:30</code> , IBM Tivoli Storage Manager displays image objects with a time of 5:30 or later on the end date you specify. |

### TYPE

Specifies the type of NDMP backup images for which you want to display information. The default value for this parameter is `BACKUPIIMAGE`. Other image types represent backup methods that might be specific to a particular file server. Possible values are:

#### **BACKUPIimage**

Specifies that the output should show only the standard NAS base and differential images. This is the default value for this parameter.

#### **SNAPMirror**

Specifies whether to display information about NetApp SnapMirror images. SnapMirror images are block-level full-backup images of a file system. A SnapMirror image can only be restored to a file system that has been prepared as a SnapMirror target volume. Refer to the documentation that came with your NetApp file server for more information. This parameter is valid for NetApp and IBM N-Series file servers only.

### Example:

Issue the `QUERY NASBACKUP` command to display information about a node, `nas1`, and a filesystem, `/vol/vol1`.

```
query nasbackup nas1 /vol/vol1
```

| Node Name | Filespace Name | Object Type (MB)   | Object Size (MB) | Creation Date Contents | Has Table of Contents (TOC) | Mgmt Class Name | Image Storage Pool Name |
|-----------|----------------|--------------------|------------------|------------------------|-----------------------------|-----------------|-------------------------|
| NAS1      | vol/vol1       | Full image         | 1050.5           | 10/22/2002 10:50:57    | YES                         | DEFAULT         | NASBACKUPS              |
| NAS1      | vol/vol1       | Differential image | 9.1              | 10/22/2002 11:03:21    | YES                         | DEFAULT         | NASBACKUPS              |
| NAS1      | vol/vol1       | Full image         | 1050.5           | 10/22/2006 10:43:00    | YES                         | STANDARD        | FILEPOOL                |
| NAS1      | vol/vol1       | Differential image | 9.1              | 10/25/2006 11:53:21    | YES                         | STANDARD        | FILEPOOL                |

### Example:

Issue the QUERY NASBACKUP command to display information about all NetApp SnapMirror to Tape images for a node, nas2, and a filesystem, /vol/vol2.

```
query nasbackup nas2 /vol/vol2 type=snapmirror
```

| Node Name | Filespace Name | Object Type | Object Size (MB) | Creation Date       | Mgmt Class Name | Image Storage Pool Name |
|-----------|----------------|-------------|------------------|---------------------|-----------------|-------------------------|
| NAS2      | vol/vol2       | SnapMirror  | 1050.5           | 04/02/2008 10:50:57 | STANDARD        | MYPPOOL                 |
| NAS2      | vol/vol2       | SnapMirror  | 1450.5           | 04/02/2008 11:03:21 | STANDARD        | MYPPOOL                 |

### Related commands

Table 232. Commands related to QUERY NASBACKUP

| Command  | Description  |
|--|--|
| BACKUP NODE  | Backs up a network-attached storage (NAS) node.                            |
| BACKUP NAS (Tivoli Storage Manager client command) | Creates a backup of NAS node data.   |
| QUERY TOC  | Displays details about the table of contents for a specified backup image. |
| RESTORE NODE                                       | Restores a network-attached storage (NAS) node.                            |

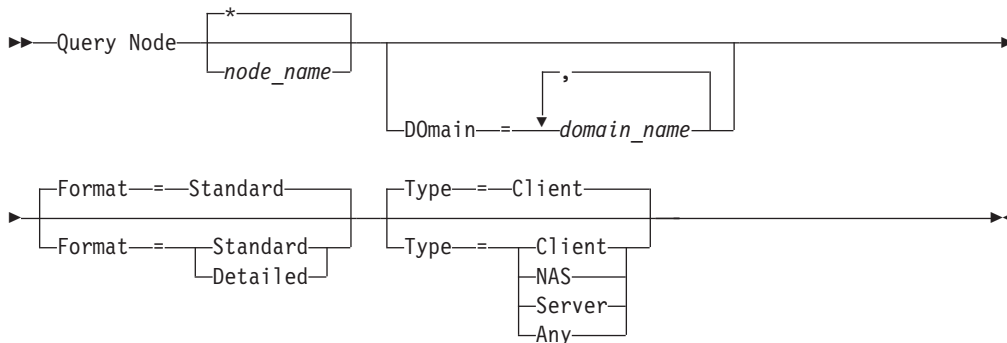
## QUERY NODE (Query nodes)

Use this command to display information about one or more registered nodes.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies the name of the client node to be queried. You can use wildcard characters 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. The parameter is optional.

#### **D0main**

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. Separate the items in the list by commas, with no intervening spaces. You can use wildcard characters to specify a domain. All clients assigned to a matching domain are displayed. If you do not specify a value for this parameter, all policy domains are included in the query.

#### **Format**

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.

#### **Type**

Specifies the type of node to include in the query results. The parameter is optional. The default value is CLIENT. Possible values are:

##### **Any**

Specifies any type of node.

##### **Client**

Specifies client nodes that are backup-archive clients, Tivoli Storage Manager for Space Management clients, or application clients.

### NAS

Specifies NAS nodes.

### Server

Specifies client nodes that are other Tivoli Storage Manager servers.

## Example: Display information on registered client nodes

Display information on all registered client nodes.

query node

| Node Name | Platform | Policy Domain Name | Days Since Last Access | Days Since Password Set | Locked? |
|-----------|----------|--------------------|------------------------|-------------------------|---------|
| CLIENT1   | AIX      | STANDARD           | 6                      | 6                       | No      |
| GEORGE    | AIX      | STANDARD           | 1                      | 1                       | No      |
| JANET     | AIX      | STANDARD           | 1                      | 1                       | No      |
| JARED     | Linux86  | STANDARD           | 1                      | 1                       | No      |
| JOE2      | Mac      | STANDARD           | <1                     | <1                      | No      |
| TOMC      | WinNT    | STANDARD           | 1                      | 1                       | No      |

## Example: Displayed detailed information about a client node

Display complete information on the client node named Joe.

query node joe format=detailed

```

Node Name: JOE
Platform: WinNT
Client OS Level: 4.00
Client Version: Version 5, Release 2, Level 0.0
Policy Domain Name: STANDARD
Last Access Date/Time: 09/24/2002 18:55:46
Days Since Last Access: 6
Password Set Date/Time: 09/24/2002 18:26:43
Days Since Password Set: 6
Invalid Sign-on Count: 0
Locked?: No
Contact:
Compression: Client
Archive Delete Allowed?: Yes
Backup Delete Allowed?: No
Registration Date/Time: 09/24/2002 18:26:43
Registering Administrator: SERVER_CONSOLE
Last Communication Method Used: Tcp/Ip
Bytes Received Last Session: 108,731
Bytes Sent Last Session: 698
Duration of Last Session (sec): 0.00
Pct. Idle Wait Last Session: 0.00
Pct. Comm. Wait Last Session: 0.00
Pct. Media Wait Last Session: 0.00
Optionset:
URL: http://joe.host.name:1581
Node Type: Client
Password Expiration Period: 60
Keep Mount Point?: No
Maximum Mount Points Allowed: 2
Auto Filespace Rename: No
Validate Protocol: No
TCP/IP Name:
TCP/IP Address: 9.11.153.39
Globally Unique ID: 11.9c.54.e0.8a.b5.11.d6.b3.c3.00.06.29.45.c1
Transaction Group Max: 0
Data Write Path: ANY
Data Read Path: ANY
Session Initiation: ClientOrServer
High-level Address:
Low-level Address: 1501
Collocation Group Name:
Proxynode Target:
Proxynode Agent:
Node Groups:

```

## Field descriptions

### Node Name

The name of the client node.

### Platform

The client node's platform (operating system) as of the last time that the client node contacted the server. A question mark (?) is displayed until the client node first accesses the server and reports its platform type.

### Client OS Level

The level of the client's operating system as of the last time that the client node contacted the server.

### Client Version

The version of the client that is installed on the client node.

This field does not apply to NAS nodes.

### Policy Domain Name

The client node's assigned policy domain.



**Last Access Date/Time**

The last date and time that the client node accessed the server.

**Days Since Last Access**

The number of days that have elapsed since the last time that the client node accessed the server.

**Password Set Date/Time**

The date and time that the client node's password was set.

**Days Since Password Set**

The number of days that have elapsed since the client node's password was set.

**Invalid Sign-on Count**

The number of invalid sign-on attempts that have been made since the last successful sign-on. This count can only be non-zero when the invalid password limit (SET INVALIDPWLIMIT) is greater than zero. When the number of invalid attempts equals the limit set by the SET INVALIDPWLIMIT command, the node is locked out of the system.

**Locked?**

Whether the client node is locked out of Tivoli Storage Manager.

**Contact**

Any contact information for the client node.

**Compression**

Whether compression is enabled on the client node.

This field does not apply to NAS nodes.

**Archive Delete Allowed?**

Whether the client node can delete its own archive files.

**Backup Delete Allowed?**

Whether the client node can delete its own backup files.

**Registration Date/Time**

The date and time that the client node was registered.

**Registering Administrator**

The name of the administrator that registered the client node.

**Last Communication Method Used**

The communication method that was last used by the client node to contact the server.

**Bytes Received Last Session**

The number of bytes received by the server during the last client node session.

This field does not apply to NAS nodes.

**Bytes Sent Last Session**

The number of bytes sent to the client node.

This field does not apply to NAS nodes.

**Duration of Last Session (sec)**

How long the most recent client node session lasted.

This field does not apply to NAS nodes.

**Pct. Idle Wait Last Session**

The percentage of the total session time that the client was not performing any functions.

This field does not apply to NAS nodes.

**Pct. Comm. Wait Last Session**

The percentage of the total session time that the client waited for a communication response from the server.

This field does not apply to NAS nodes.

**Pct. Media Wait Last Session**

The percentage of the total session time that the client waited for a removable volume to be mounted.

This field does not apply to NAS nodes.

**Optionset**

The name of the client option set.

**URL** The client's Uniform Resource Locator (URL) address that the administrator can use to contact the client from the Administration Center.

**Node Type**

The type of client node. The value is one of the following:

- Client (a backup-archive client, a Tivoli Storage Manager for Space Management client, or an application client)
- Server (a Tivoli Storage Manager server)
- NAS (a NAS file server)

**Password Expiration Period**

The client node's password expiration period.

**Keep Mount Point?**

Whether the client node has the ability to retain a mount point during a session.

**Maximum Mount Points Allowed**

The mount points that a client node is allowed to use on the server for Tivoli Storage Manager for Space Management migration and for backup and archive operations. This parameter does not apply to nodes with a type of NAS or SERVER. If a client node was registered to a server at Version 3.7 or later, the value will be 0-999, depending on the value that is set with the MAXNUMMP parameter of the REGISTER NODE command. If the client node was registered under previous versions of the server and the MAXNUMMP parameter was not explicitly set using the UPDATE NODE command, then the value is set to NOLIMIT. The MAXNUMMP value is not evaluated or enforced during client data read operations such as restore, retrieve, and Tivoli Storage Manager for Space Management recall. However, mount points in use for data read operations are evaluated against attempted concurrent data store operations for the same client node and may prevent the data store operations from being able to acquire mount points.

**Auto Filespace Rename**

Whether Tivoli Storage Manager prompts the client to rename file spaces when the client system upgrades to a client that supports Unicode. This field is valid only for client systems that use Windows, Macintosh OS X, or NetWare operating systems.

**Validate Protocol**

Whether the client has data validation enabled. If the client has data validation enabled, this field specifies whether Tivoli Storage Manager validates only the file data or all data, which includes file metadata. You can enable data validation by using the REGISTER NODE or UPDATE NODE commands.

**TCP/IP Name**

The host name of the client node as of the last time that the client node contacted the server. The field is blank if the client software does not support reporting this information to the server.

**TCP/IP Address**

The TCP/IP address of the client node as of the last time that the client node contacted the server. The field is blank if the client software does not support reporting this information to the server.

**Globally Unique ID**

The globally unique identifier (GUID) as of the last time that the client node contacted the server. This GUID identifies the host machine on which the node resides.

**Transaction Group Max**

Specifies the number of files per transaction commit that are transferred between a client and a server. Client performance may be improved by using a larger value for this option.

**Data Write Path**

Specifies the transfer path used when the client sends data to the server, storage agent, or both during storage operations such as backup or archive. If a path is unavailable, the node cannot send any data.

Data transfer path options are: ANY, LAN, or LAN-free.

**Data Read Path**

Specifies the transfer path used when the server, storage agent, or both read data for a client, during operations such as restore or retrieve. If a path is unavailable, data cannot be read.

Data transfer path options are: ANY, LAN, or LAN-free.

**Session Initiation**

Controls whether the server or client initiates sessions. The two options are:

- ClientOrServer
- Serveronly

**High-level Address**

Specifies the client IP address that the server contacts to initiate scheduled events when SESSIONINITIATION is set to SERVERONLY.

**Low-level Address**

Specifies the client port number on which the client listens for sessions from the server when SESSIONINITIATION is set to SERVERONLY.

**Collocation Group Name**

Specifies the name of the collocation group to which a node belongs. If a node does not belong to a collocation group, this field is blank.

## QUERY NODE

### Proxynode Target

Specifies which nodes are proxy nodes (agents) for other nodes, in a space separated list. If there are no nodes in that type of association, this field is blank.

### Proxynode Agent

Specifies the originating (target) node name for a proxy node session, in a space separated list. If there are no nodes in that type of association, this field is blank.

### Node Groups

Specifies the name of the node group to which a node belongs. If a node does not belong to a node group, this field is blank.

## Related commands

Table 233. Commands related to QUERY NODE

| Command            | Description   |
|--------------------|---|
| LOCK NODE          | Prevents a client from accessing the server.  |
| QUERY ADMIN        | Displays information about one or more IBM Tivoli Storage Manager administrators.       |
| QUERY STATUS       | Displays the settings of server parameters, such as those selected by the SET commands. |
| REGISTER NODE      | Defines a client to the server and sets options for that user.                          |
| REMOVE NODE        | Removes a client from the list of registered nodes for a specific policy domain.        |
| RENAME NODE        | Changes the name for a client node.   |
| RESET PASSEXP      | Resets the password expiration for nodes or administrators.                             |
| SET INVALIDPWLIMIT | Sets the number of invalid logon attempts before a node is locked.                      |
| SET MINPWLENGTH    | Sets the minimum length for client passwords.   |
| SET PASSEXP        | Specifies the number of days after which a password is expired and must be changed.     |
| UNLOCK NODE        | Enables a locked user in a specific policy domain to access the server.                 |
| UPDATE NODE        | Changes the attributes associated with a client node.                                   |

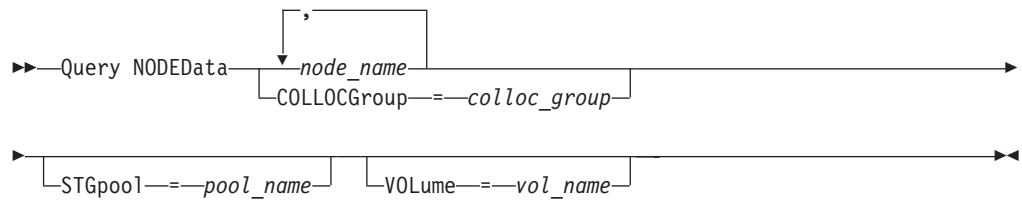
## QUERY NODEDATA (Query client data in volumes)

Use this command to display information about the data for one or more nodes in a sequential access storage pool. QUERY NODEDATA displays the name of the volume on which a node's data is written and the amount of space occupied by the data on that volume. This information is useful when determining how to group nodes into collocated storage pools.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies the name of the client node for which you want to locate data. You can specify one or more names. If you specify multiple names, separate the names with commas; do not use intervening spaces. You can also use wildcard characters when specifying multiple names. You must specify either a node name or collocation group name, but not both.

#### COLLOCGroup

Specifies the name of the collocation group for which you want to locate data. You must specify either a node name or collocation group name, but not both.

#### STGpool

Specifies the name of the sequential storage pool to query. This parameter is optional. You can use wildcard characters to specify the names. If a wildcard matches the name of a disk storage pool, the name of the disk storage pool will be ignored. If you do not specify a value for this parameter, all sequential storage pools are queried.

#### VOLume

Specifies the volume that contains the data. This parameter is optional. You can use wildcard characters to specify multiple names. If you do not specify a value for this parameter, all volumes in the storage pool are queried.

### Example: Use wildcards to display node data for a sequential access storage pool

Display information about where node data is stored in a sequential storage pool. Use a wildcard character to indicate node names. See "Field descriptions" on page 720 for field descriptions.

```
query nodedata e*
```

## QUERY NODEDATA

| Node Name | Volume Name                | Storage Pool Name | Physical Space Occupied (MB) |
|-----------|----------------------------|-------------------|------------------------------|
| -----     | -----                      | -----             | -----                        |
| EDU_J2    | E:\tsm\server\00000117.BFS | EDU512            | 0.01                         |
| EDU_J2    | E:\tsm\server\00000122.BFS | EDU319            | 0.01                         |
| EDU_J3    | E:\tsm\server\00000116.BFS | EDU512            | 0.01                         |
| EDU_J3    | E:\tsm\server\00000120.BFS | EDU319            | 0.01                         |
| EDU_J7    | E:\tsm\server\00000118.BFS | EDU512            | 0.04                         |
| EDU_J7    | E:\tsm\server\00000123.BFS | EDU319            | 0.04                         |
| EDU_JJ1   | E:\tsm\server\00000116.BFS | EDU512            | 0.01                         |
| EDU_JJ1   | E:\tsm\server\00000121.BFS | EDU512            | 0.01                         |

### Example: Display node data information for a specific collocation group

Display information about the location of node data in a sequential storage pool for a particular collocation group. In this example, nodes EDU\_J3 and EDU\_JJ1 are the only members that belong to collocation group, grp1, and have data in a sequential access storage pool. See “Field descriptions” for field descriptions.

query nodedata collocgroup=grp1

| Node Name | Volume Name                | Storage Pool Name | Physical Space Occupied (MB) |
|-----------|----------------------------|-------------------|------------------------------|
| -----     | -----                      | -----             | -----                        |
| EDU_J3    | E:\tsm\server\00000116.BFS | EDU512            | 0.01                         |
| EDU_J3    | E:\tsm\server\00000120.BFS | EDU319            | 0.01                         |
| EDU_JJ1   | E:\tsm\server\00000116.BFS | EDU512            | 0.01                         |
| EDU_JJ1   | E:\tsm\server\00000121.BFS | EDU512            | 0.01                         |

### Field descriptions

#### Node Name

Specifies the name of the node.

#### Volume Name

Specifies the name of the volume that contains the node data.

#### Storage Pool Name

Specifies the name of the storage pool in which the volume is located.

#### Physical Space Occupied (MB)

Specifies the amount of physical space occupied by the node's data.  
Physical space includes empty space within aggregates, from which files may have been deleted or expired.

### Related commands

Table 234. Commands related to QUERY NODEDATA

| Command             | Description   |
|---------------------|---|
| DEFINE COLLOCGROUP  | Defines a collocation group.  |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                            |
| DEFINE STGPOOL      | Defines a storage pool as a named collection of server storage media. |
| DELETE COLLOCGROUP  | Deletes a collocation group.  |

Table 234. Commands related to QUERY NODEDATA (continued)

| Command             | Description  |
|---------------------|--|
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                  |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces.    |
| QUERY COLLOCGROUP   | Displays information about collocation groups.                                   |
| QUERY NODE          | Displays partial or complete information about one or more clients.              |
| QUERY STGPOOL       | Displays information about storage pools.  |
| REMOVE NODE         | Removes a client from the list of registered nodes for a specific policy domain. |
| UPDATE COLLOCGROUP  | Updates the description of a collocation group.                                  |
| UPDATE STGPOOL      | Changes the attributes of a storage pool.  |

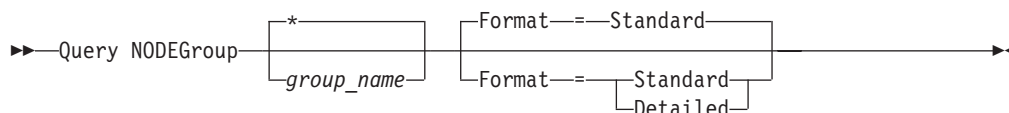
### QUERY NODEGROUP (Query a node group)

Use this command to display the node groups defined on the server.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *group\_name*

Specifies the name of the node group to display. To specify multiple names, use a wildcard character. This parameter is optional. The default is to display all node groups.

##### Format

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.

##### Detailed

Specifies that complete information is displayed. To display the members of the node group, you must specify FORMAT=DETAILED.

#### Example: List node groups on the server

Display the node groups defined on the server. See “Field descriptions” on page 723 for field descriptions.

```
query nodegroup
```

| Node Group Name | Node Group Description |
|-----------------|------------------------|
| DEPT_ED         | Education department   |
| GROUP1          | Low cap client nodes.  |

#### Example: Display detailed node group information

Display complete information about all node groups and determine which client nodes belong to which node groups. See “Field descriptions” on page 723 for field descriptions.

```
query nodegroup format=detailed
```



```

Node Group Name: DEPT_ED
Node Group Description: Education department
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2006 10:59:03
Node Group Member(s): EDU_1 EDU_7

Node Group Name: GROUP1
Node Group Description: Low cap client nodes.
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2006 10:59:16
Node Group Member(s): CHESTER REX NOAH JARED

```

## Field descriptions

### Node Group Name

The name of the node group.

### Node Group Description

The description for the node group.

### Last Update by (administrator)

The name of the administrator that defined or most recently updated the node group.

### Last Update Date/Time

The date and time that an administrator defined or most recently updated the node group.

### Node Group Member(s)

The members of the node group.

## Related commands

Table 235. Commands related to QUERY NODEGROUP

| Command                | Description   |
|------------------------|---|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP       | Defines a group of nodes.                               |
| DEFINE NODEGROUPMEMBER | Adds a client node to a node group.                     |
| DELETE BACKUPSET       | Deletes a backup set.                                   |
| DELETE NODEGROUP       | Deletes a node group.                                   |
| DELETE NODEGROUPMEMBER | Deletes a client node from a node group.                |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.              |
| QUERY BACKUPSET        | Displays backup sets.                                   |
| UPDATE BACKUPSET       | Updates a retention value associated with a backup set. |
| UPDATE NODEGROUP       | Updates the description of a node group.                |

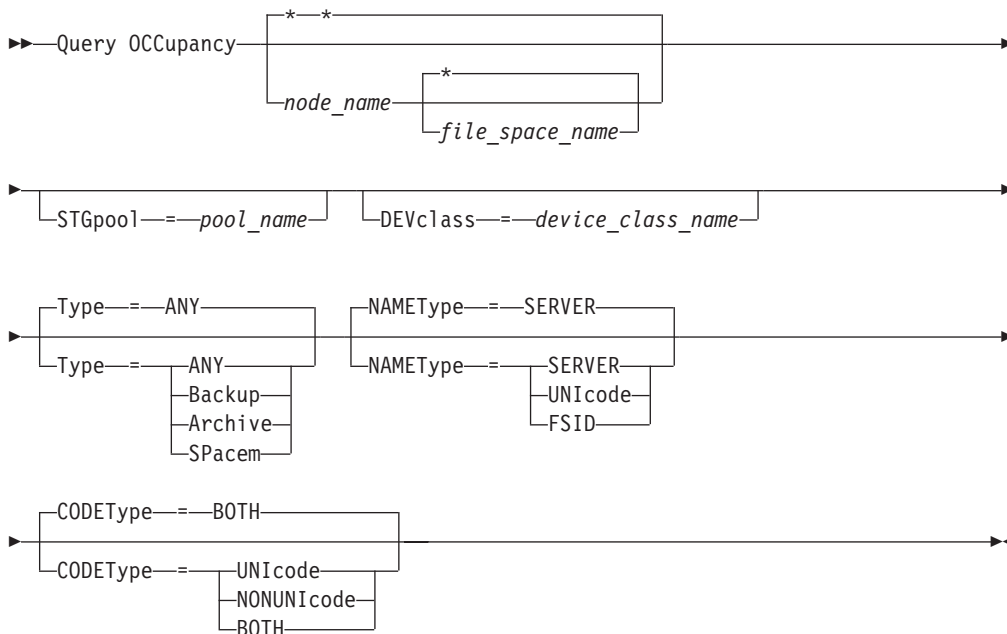
## QUERY OCCUPANCY (Query client file spaces in storage pools)

Use this command to show where client file spaces are stored and how much space they occupy.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies the node that owns the file spaces that you want to locate. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a value for this parameter, all nodes are queried.

#### *file\_space\_name*

Specifies the file space that you want to locate. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a value for this parameter, all file spaces are queried. You must specify a node name if you specify a file space name.

For a server that has clients with Unicode support, you may need to have the server convert the file space name that you enter. For example, you may need to have the server convert the name you enter from the server's code page to Unicode. See the NAME TYPE parameter for details. If you do not specify a file space name or specify only a single wildcard character for the name, you can use the CODE TYPE parameter to limit the operation to Unicode file spaces or non-Unicode file spaces.

#### **STGpool**

Specifies the storage pool to query for files from the specified file space. This

parameter is optional. You can use wildcard characters to specify names. If you do not specify a value for this parameter, all storage pools are queried.

### **DEVclass**

Specifies the device class associated with the devices where the file spaces are stored. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a value for this parameter, storage pools associated with any device class are queried.

### **Type**

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 Tivoli Storage Manager for Space Management clients.

#### **Backup**

Specifies that backup files are queried.

#### **Archive**

Specifies that archive files are queried.

#### **SPacem**

Specifies that space-managed files (files that were migrated by a Tivoli Storage Manager for Space Management client) are queried.

### **NAMEType**

Specifies how you want the server to interpret the file space names that you enter. This parameter is useful when the server has clients with Unicode support. A backup-archive client with Unicode support is currently available only for Windows, Macintosh OS 9, Macintosh OS X, and NetWare. Use this parameter only when you specify a partly or fully qualified file space name.

The default value is SERVER. Possible values are:

#### **SERVER**

The server uses the server's code page to interpret the file space names.

#### **UNICODE**

The server converts the file space names from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the names and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

#### **FSID**

The server interprets the file space names as their file space IDs (FSIDs).

### **CODEType**

Specifies how you want the server to interpret the file space names that you enter. Use this parameter only when you enter a single wildcard character for the file space name or when you do not specify any file space name.

The default value is BOTH, which means that the file spaces are included regardless of code page type. Possible values are:

#### **UNICODE**

Include only file spaces that are Unicode enabled.

#### **NONUNICODE**

Include only file spaces that are not Unicode enabled.

## QUERY OCCUPANCY

### BOTH

Include file spaces regardless of code page type.

### Example: Display file spaces assigned to a specific node

Display information on where all file spaces assigned to the node named DAISY are stored. See “Field descriptions” for field descriptions.

```
query occupancy daisy
```

| Node Name | Type | Filespace Name | FSID | Storage Pool Name | Number of Files | Physical Space Occupied (MB) | Logical Space Occupied (MB) |
|-----------|------|----------------|------|-------------------|-----------------|------------------------------|-----------------------------|
| DAISY     | Bkup | DRIVED         | 1    | COPYFILE          | 38              | 0.45                         | 0.42                        |

### Example: Display file spaces assigned to a specific node with a backup file type

Display information on the file spaces that belong to the node WAYNE, and that have a backup file type. See “Field descriptions” for field descriptions.

```
query occupancy wayne type=backup
```

| Node Name | Type | Filespace Name | FSID | Storage Pool Name | Number of Files | Physical Space Occupied (MB) | Logical Space Occupied (MB) |
|-----------|------|----------------|------|-------------------|-----------------|------------------------------|-----------------------------|
| WAYNE     | Bkup | DWG1           | 1    | BACKUPPOOL1       | 2,330           | 53.19                        | 50.01                       |
| WAYNE     | Bkup | OS2C           | 2    | BACKUPPOOL1       | 1,554           | 32.00                        | 31.30                       |

## Field descriptions

### Node Name

The node that owns the file space.

**Type** The type of data. Possible values are:

**Arch** Data that has been archived.

**Bkup** Data that has been backed up.

**Spmg** Data that has been migrated from a Tivoli Storage Manager for Space Management client.

### Filespace Name

The name of the file space that belongs to the node. Where the name of the file space is *(archive)*, that data was archived before collocation by file space was added to the IBM Tivoli Storage Manager product.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These

characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

## Storage Pool Name

The storage pool where the file space currently resides.

## Number of Files

The number of logical files that belong to the file space and are stored in this storage pool.

## Physical Space Occupied (MB)

The amount of physical space occupied by the file space. Physical space includes empty space within aggregates, from which files may have been deleted or expired.

**Tip:** This field does not display a value for storage pools that are set up for deduplication. If you turn off deduplication for a storage pool, a value for physical occupancy is not displayed until the storage pool is empty of deduplicated files.

## Logical Space Occupied (MB)

The amount of space occupied by logical files in the file space. Logical space is the space actually used to store files, excluding empty space within aggregates.

**FSID** The file space ID (FSID) for the file space. The server assigns a unique FSID when a file space is first stored on the server.

## Related commands

Table 236. Commands related to QUERY OCCUPANCY

| Command          | Description   |
|------------------|---|
| DELETE FILESPACE | Deletes data associated with client's file spaces.                      |
| QUERY FILESPACE  | Displays information about data in file spaces that belong to a client. |
| QUERY NODE       | Displays partial or complete information about one or more clients.     |

## QUERY OPTION (Query server options)

Use this command to display information about server options.

Change server options by editing the server options file or by issuing the SETOPT command. If you edit the server options file, you must restart the server before any changes take effect. Any changes you make by issuing the SETOPT command take effect immediately.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

*optionname*

Specifies the name of an option in the server options file. This parameter is optional. You can use wildcard characters to specify this name. All matching server options display. If you do not specify this parameter, information on all options displays.

### Example: Display all server options

Display general information about all server options. The output lists all options with their specified values.

```
query option
```

### Example: Display options settings using a wildcard

View the option settings for all options that begin with L.

```
query option l*
```

| Server Option | Option Setting |
|---------------|----------------|
| -----         | -----          |
| Language      | AMENG          |

### Related commands

Table 237. Commands related to QUERY OPTION

| Command | Description   |
|---------|---|
| SETOPT  | Updates a server option without stopping and restarting the server. |

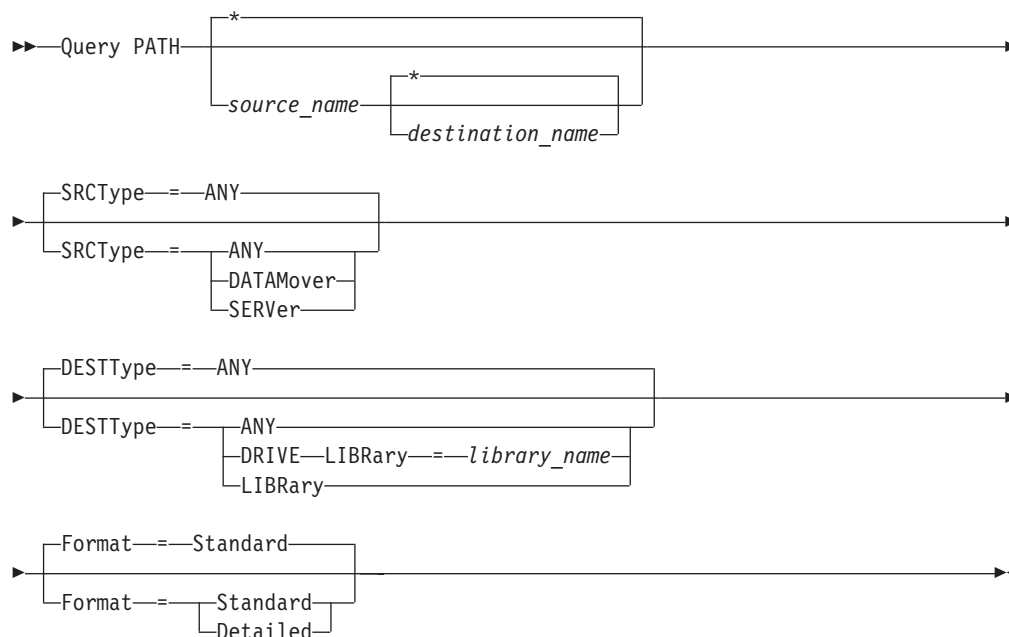
## QUERY PATH (Display a path definition)

Use this command to display the path between a source and a destination.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *source\_name*

Specifies the name of a source for which to display paths. This parameter is optional. You can specify wildcard characters. The default is to display paths for all sources.

A source is a data mover, a server, or a storage agent.

#### *destination\_name*

Specifies the name of a destination for which to display paths. This parameter is optional. You can specify wildcard characters. The default is to display paths for all destinations.

#### **SRCType**

Specifies the type of the source. This parameter is optional. The default is to display paths for all source types. Possible values are:

##### **ANY**

Specifies to display paths with any source type.

##### **DATAMover**

Specifies to only display paths with the DATAMOVER source type.

##### **SERVer**

Specifies to only display paths with the SERVER source type. (A source that has a source type of SERVER is a storage agent.)

### DESTType

Specifies the type of the destination. This parameter is optional. The default is to display paths for all destination types. Possible values are:

#### ANY

Specifies to display paths with any destination type.

#### DRive

Specifies to display only paths with the DRIVE destination type. When the destination type is a drive, you must specify the library name. You can refine which paths are displayed by entering a name in the LIBRARY parameter.

#### LIBRARY

Specifies that only paths with destination type LIBRARY display.

### LIBRARY

Specifies the name of the library to which the drive belongs. This parameter is required when the destination type is a drive (DESTTYPE=DRIVE).

### Format

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

#### Standard

Specifies that partial information is displayed.

#### Detailed

Specifies that complete information is displayed.

## Example: Display summary path information

Display information about paths for the source NETAPP1. See “Field descriptions” on page 731 for field descriptions.

```
query path netapp1
```

| Source Name | Source Type | Destination Name | Destination Type | Online |
|-------------|-------------|------------------|------------------|--------|
| NETAPP1     | DATAMOVER   | DRIVE1           | DRIVE            | Yes    |
| NETAPP1     | DATAMOVER   | NASLIB           | LIBRARY          | Yes    |

## Example: Display detailed path information

Display detailed information about paths for the source NETAPP1. See “Field descriptions” on page 731 for field descriptions.

```
query path netapp1 format=detailed
```



```

Source Name: NETAPP1
Source Type: DATAMOVER
Destination Name: NASLIB
Destination Type: LIBRARY
Library:
Device: mc0
Directory:
On-Line: Yes
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 06/21/2001 20:52:56

Source Name: NETAPP1
Source Type: DATAMOVER
Destination Name: DRIVE1
Destination Type: DRIVE
Library: NASLIB
Device: rst01
Directory:
On-Line: Yes
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 06/21/2001 20:55:23

```

## Field descriptions

### Source Name

The name of the source.

### Destination Name

The name of the destination.

### Source Type

The type of the source.

### Destination Type

The type of the destination.

### Library

The name of the library that contains the drive that is the destination. This field will be blank if the destination type is library. The library name is in destination name field when the destination is a library.

### Node Name

The name of the device that is the destination. Not applicable to MVS.

### Device

The name of the device that is the destination. Not applicable to MVS.

### Directory

Specifies the directory location of a file on the source.

### LUN

Specifies the logical unit name through which the disk can be accessed by the source.

### Online

Whether the path is online and available for use.

### Last Update by (administrator)

The ID of the administrator who performed the last update.

### Last Update Date/Time

The date and time when the last update occurred.

### Related commands

*Table 238. Commands related to QUERY PATH*

| Command     | Description                                    |
|-------------|--|
| DEFINE PATH | Defines a path from a source to a destination. |
| DELETE PATH | Deletes a path from a source to a destination. |
| UPDATE PATH | Changes the attributes associated with a path. |

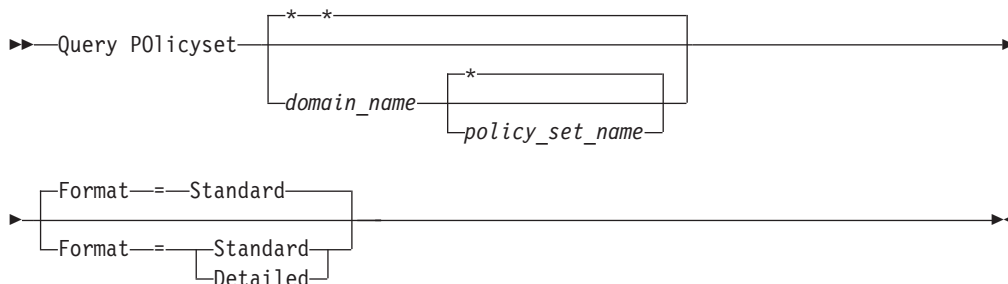
## QUERY POLICYSET (Query a policy set)

Use this command to display information about one or more policy sets.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *domain\_name*

Specifies the policy domain associated with the policy set to query. This parameter is optional. You can use wildcard characters to specify names. 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.

#### *policy\_set\_name*

Specifies the policy set to query. This parameter is optional. You can use wildcard characters to specify names. If you do not specify either ACTIVE or a policy set name, all policy sets are queried.

#### Format

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.

##### **Detailed**

Specifies that complete information is displayed.

### Example: List policy sets for all policy domains

Query all policy sets for all policy domains. Create the output in standard format. See “Field descriptions” on page 734 for field descriptions.

```
query policyset
```

## QUERY POLICYSET

| Policy Domain Name | Policy Set Name | Default Mgmt Class Name | Description                   |
|--------------------|-----------------|-------------------------|-------------------------------|
| -----              | -----           | -----                   | -----                         |
| EMPLOYEE-RECORDS   | ACTIVE          | ACTIVEFILES             | Personnel Department          |
| EMPLOYEE-RECORDS   | HOLIDAY         | ACTIVEFILES             | Personnel Department          |
| EMPLOYEE-RECORDS   | VACATION        | ACTIVEFILES             | Personnel Department          |
| PROG1              | SUMMER          |                         | Programming Group Policies    |
| PROG2              | SUMMER          |                         | Programming Group Policies    |
| STANDARD           | ACTIVE          | STANDARD                | Installed default policy set. |
| STANDARD           | STANDARD        | STANDARD                | Installed default policy set. |

### Example: Displayed detailed information about a specific policy set

Query the VACATION policy set that is in the EMPLOYEE\_RECORDS policy domain. Create the output in detailed format. See “Field descriptions” for field descriptions.

```
query policyset employee_records vacation
format=detailed
```

```
Policy Domain Name: EMPLOYEE_RECORDS
Policy Set Name: VACATION
Default Mgmt Class Name: ACTIVEFILES
Description: Personnel Department
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 05/31/1998 13:15:50
Managing profile: ADSM_INFO
Changes Pending: Yes
```

### Field descriptions

#### Policy Domain Name

The name of the policy domain.

#### Policy Set Name

The name of the policy set.

#### Default Mgmt Class Name

The management class assigned as the default for the policy set.

#### Description

The description of the policy set.

#### Last Update by (administrator)

The name of the administrator or server that most recently updated the policy set. If this field contains \$\$CONFIG\_MANAGER\$\$, the policy set is associated with a domain that is managed by the configuration manager.

#### Last Update Date/Time

The date and time when the policy set was most recently defined or updated.

### Managing Profile

The profile or profiles that manage the domain to which this policy set belongs.

### Changes Pending

Whether or not changes are being made but not activated. Once the changes are activated, the field resets to No.

### Related commands

Table 239. Commands related to QUERY POLICYSET

| Command            | Description  |
|--------------------|--|
| ACTIVATE POLICYSET | Validates and activates a policy set.  |
| COPY POLICYSET     | Creates a copy of a policy set.  |
| DEFINE POLICYSET   | Defines a policy set within the specified policy domain.   |
| DELETE POLICYSET   | Deletes a policy set, including its management classes and copy groups, from a policy domain.        |
| QUERY DOMAIN       | Displays information about policy domains.   |
| UPDATE POLICYSET   | Changes the description of a policy set.   |
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

### QUERY PROCESS (Query one or more server processes)

Use this command to display information on active background processes.

Use the CANCEL PROCESS command to cancel background processes.

#### Privilege class

Any administrator can issue this command.

#### Syntax

```

▶▶ Query Process _____▶▶
      |
      | process_number
      |

```

#### Parameters

*process\_number*

Specifies the number of the background process to be queried. This parameter is optional. If not specified, information on all background processes is displayed.

#### Example: Query a single background process

Display information about background process 202. See “Field descriptions” on page 737 for field descriptions.

```
query process 202
```

| Process Number | Process Description | Status   |
|----------------|---------------------|--|
| 202            | EXPORT SERVER       | ANR0NNNI EXPORT<br>Identifier MYEXPORTSERVER<br>ANR0648I Have copied the<br>following: 8 Domains 2<br>Policy Sets 10 Management<br>Classes 4 Copy Groups 1<br>Administrators 746 Bytes<br>(0 errors have been<br>detected) Current input<br>volume(s): C:\BUILD\540\<br>GA\BUILD\NT\I386\DEBUG\<br>-00000014.BFS,(6 Seconds) |

#### Example: Query all background processes

Display information about all background processes. See “Field descriptions” on page 737 for field descriptions.

```
query process
```

| Process<br>Number | Process<br>Description | Status  |
|-------------------|------------------------|---|
| 304               | IDENTIFY DUPLICATES    | Storage Pool<br>FILEPOOL, Volume<br>/tsmpool2/00006664.<br>BFS, Files Processed:<br>2000, Duplicate Extents<br>Found: 344, Duplicate<br>Bytes Found: 3,238,123,<br>Current Physical File<br>(bytes): 2,626,676,296.<br>Status: Processing |
| 284               | IDENTIFY DUPLICATES    | Storage Pool<br>FILEPOOL, Volume<br>/tsmpool2/00006666.<br>BFS, Files Processed:<br>2000, Duplicate Extents<br>Found: 344, Duplicate<br>Bytes Found: 3,238,123,<br>Current Physical File<br>(bytes): None.<br>Status: Idle                |

## Field descriptions

### 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 240. Command related to QUERY PROCESS

| Command        | Description   |
|----------------|---|
| CANCEL EXPORT  | Deletes a suspended export operation                                    |
| CANCEL PROCESS | Cancels a background server process.                                    |
| QUERY EXPORT   | Displays the export operations that are currently running or suspended. |
| RESTART EXPORT | Restarts a suspended export operation.                                  |
| SUSPEND EXPORT | Suspends a running export operation.                                    |

## QUERY PROFILE (Query a profile)

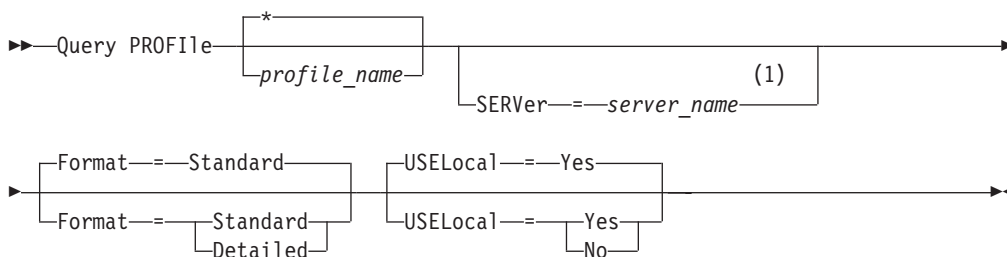
Use this command to display information about profiles and associated objects. Issue this command from a configuration manager or from a managed server. You can use this command to get profile information from any configuration manager defined to the server, even if the server does not subscribe to any profile.

If you query a locked profile from the configuration manager to which the profile belongs, complete profile information is displayed. If you query a locked profile from another server, the query displays only that the profile is locked.

### Privilege class

Any administrator can issue this command.

### Syntax



### Notes:

- 1 The server name you specify depends on the server from which you issue the command. See the description of the SERVER parameter.

### Parameters

#### *profile\_name*

Specifies the profile to display. To specify multiple names, use a wildcard character. This parameter is optional. The default is to display all profiles.

#### SERVer

Specifies the configuration manager whose profile information is displayed. The requirements for the name depends on where the query is issued:

- From a configuration manager: This parameter is optional. The default is the configuration manager's name.
- From a managed server: This parameter is optional. The default is the name of the configuration manager for this managed server.
- From a server that is neither a configuration manager nor a managed server: You must specify a name.

#### Format

Specifies whether partial or detailed information is displayed. The default is STANDARD. Possible values are:

##### Standard

Specifies that partial information is displayed.

##### Detailed

Specifies that detailed information is displayed.



USELocal

When you perform the query from a managed server, this parameter specifies whether the profile information is obtained from the configuration manager or the managed server. If the profile information does not exist on the managed server, the information is obtained from the configuration manager, regardless of the value of this parameter.

If you use this parameter on a server that is not managed by the configuration manager that owns the profile, the parameter is ignored. The default value is YES. Possible values are:

Yes

Specifies that the profile information, if available, is obtained from the managed server. The configuration manager is contacted if information is not available from the managed server.

No

Specifies that the profile information is obtained from the configuration manager even if the information is available from the managed server. This ensures that you receive current information about the profile.

Example: List profiles from a configuration manager

Display profile information from a configuration manager. See “Field descriptions” for field descriptions.

query profile

| Configuration manager | Profile name    | Locked? |
|-----------------------|-----------------|---------|
| -----                 | -----           | -----   |
| SERVER1               | DEFAULT_PROFILE | No      |
| SERVER1               | ADMIN_INFO      | No      |
| SERVER1               | EMPLOYEE        | No      |
| SERVER1               | PERSONNEL       | Yes     |

Example: Display detailed profile information for a managed server

From a managed server, display current detailed information for profile ADMIN\_INFO. See “Field descriptions” for field descriptions.

**Note:** When the profile is locked, most fields are not displayed.

query profile admin\_info  
format=detailed useLocal=no

|   |
|---|
| Configuration manager: SERVER1                      |
| Profile name: ADMIN_INFO                            |
| Locked: No  |
| Description: Distributed administrative schedules   |
| Server administrators: DENNIS EMILY ANDREA          |
| Policy domains: ADMIN RECORDS                       |
| Administrative command schedules: ** all objects ** |
| Server Command Scripts:                             |
| Client Option Sets:                                 |
| Servers:  |
| Server Groups:                                      |

Field descriptions

Configuration manager

The name of the configuration manager that owns the profile.

## QUERY PROFILE

**Profile name**

The name of the profile.

**Locked?**

Whether the profile is locked.

**Description**

The description of the profile.

**Server administrators**

The administrators that are associated with the profile.

**Policy domains**

The policy domains that are associated with the profile.

**Administrative command schedules**

The administrative schedules that are associated with the profile.

**Server Command Scripts**

The server command scripts that are associated with the profile.

**Client Option Sets**

The client option sets that are associated with the profile.

**Servers**

The servers that are associated with the profile.

**Server Groups**

The names of server groups that are associated with the profile.

### Related commands

*Table 241. Commands related to QUERY PROFILE*

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                       |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                 |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DEFINE SUBSCRIPTION    | Subscribes a managed server to a profile.                          |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                    |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                  |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |
| UNLOCK PROFILE         | Enables a locked profile to be distributed to managed servers.     |
| UPDATE PROFILE         | Changes the description of a profile.                              |

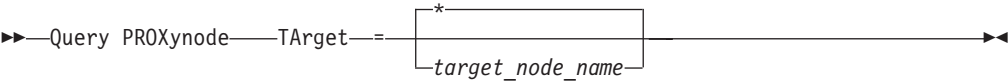
QUERY PROXYNODE (Query proxy authority for a client node)

Use this command to display client nodes with authority to act as proxy to other client nodes in the Tivoli Storage Manager server.

Privilege class

Any administrator can issue this command.

Syntax



Parameters

TArget

Specifies the name of the node targeted by the node with proxy authority. It is optional to specify a target node name. Wildcard names can be used to specify the target node name. A comma-separated list of node names is also allowed.

Example: List client nodes with proxy authority

To display all Tivoli Storage Manager client nodes with proxy authority to the target node named MYCLUSTER, issue the following command.

```
query proxynode target=mycluster
```

| Target Node | Agent Node       |
|-------------|------------------|
| -----       | -----            |
| FRED        | MOE MINIE MICKEY |
| ALPHA       | BETA GAMMA DELTA |

Related commands

Table 242. Commands related to QUERY PROXYNODE

| Command          | Description                                |
|------------------|--|
| GRANT PROXYNODE  | Grant proxy authority to an agent node.    |
| REVOKE PROXYNODE | Revoke proxy authority from an agent node. |

## QUERY RECOVERYMEDIA (Query recovery media)

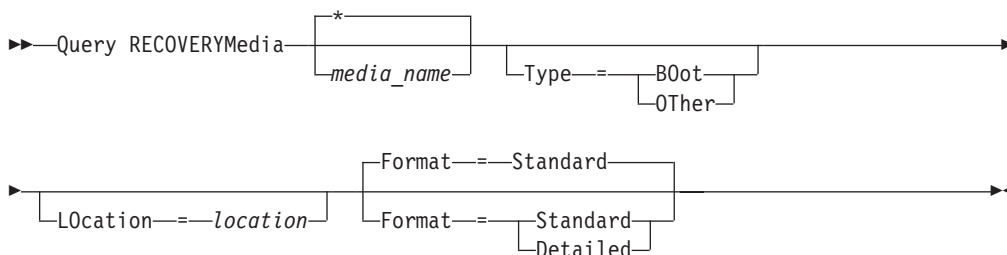
Use this command to display information about the media (for example, boot media) needed to recover a machine. Media are displayed in alphabetical order by name.

**Remember:** Tivoli Storage Manager does not use the information. It is available only to help you plan for the disaster recovery of client machines.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *media\_name*

Specifies the name of the recovery media. You can use wildcard characters to specify the name. This parameter is optional. The default is all recovery media.

#### Type

Specifies the type of media to be queried. This parameter is optional. If this parameter is not specified, all recovery media are queried. Possible values are:

##### **B0ot**

Only boot media are queried.

##### **OTher**

All media other than boot media are queried.

#### **L0cation**

Specifies the location of the recovery media to be queried. This parameter is optional. You can specify up to 255 characters. Enclose the description in quotation marks if it contains any blank characters.

#### **Format**

Specifies how the information is displayed. This parameter is optional. Possible values are:

##### **Standard**

Tivoli Storage Manager displays partial information. This is the default.

##### **Detailed**

Tivoli Storage Manager displays all information.

### Example: Display summary information for a specific recovery media

Display information for the recovery media named RECMED1. See “Field descriptions” for field descriptions.

```
query recoverymedia RECMED1
```

| Recovery Media Name | Volume Names           | Location     | Machine Name |
|---------------------|------------------------|--------------|--------------|
| -----               | -----                  | -----        | -----        |
| RECMED1             | vol1 vol2 vol3<br>vol4 | IRONMOUNTAIN | MACH1        |

### Example: Display detailed information for a specific recovery media

Display detailed information for the recovery media named RECMED1. See “Field descriptions” for field descriptions.

```
query recoverymedia RECMED1 format=detailed
```

```
Recovery Media Name: RECMED1
Type: Boot
Volume Names: vol1 vol2 vol3 vol4
Location: IRONMOUNTAIN
Description:
Product:
Product Information:
Machine Name: MACH1
```

## Field descriptions

### Recovery Media Name

The name of the recovery media.

**Type** Whether the recovery media are boot media or another type of media.  
Possible values are:

**Boot** The recovery media are boot media.

**Other** The recovery media are not boot media.

### Volume Names

The set of volumes that contain the data needed to recover machines associated with this media.

### Location

Where the recovery media is stored.

### Description

A description of the recovery media.

### Product

The product used to create the boot media.

### Product Information

Information about the product that created the boot media. This information may be needed for restoring the machine.

### Machine Name

The machines that are associated with this recovery media.

## QUERY RECOVERYMEDIA

### Related commands

*Table 243. Commands related to QUERY RECOVERYMEDIA*

| Command                      | Description                                      |
|------------------------------|--|
| DEFINE RECMEDMACHASSOCIATION | Associates recovery media with a machine.        |
| DEFINE RECOVERYMEDIA         | Defines the media required to recover a machine. |
| DELETE RECOVERYMEDIA         | Deletes recovery media.                          |
| UPDATE RECOVERYMEDIA         | Changes the attributes of recovery media.        |

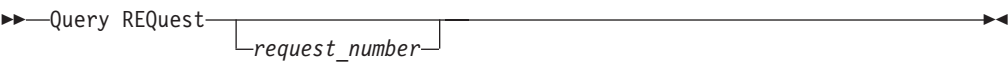
QUERY REQUEST (Query one or more pending mount requests)

Use this command to display information about one or more pending mount requests. The server makes requests for the administrator to perform an action, like inserting a tape volume in a library after a CHECKIN LIBVOL has been issued.

Privilege class

Any administrator can issue this command.

Syntax



Parameters

*request\_number*  
Specifies the identification number of the pending mount request. This parameter is optional. The default is all pending mount requests.

Example: List all pending mount requests

Display information on all pending mount requests after a CHECKIN LIBVOL has been issued.

query request

Output for a Manual Library

```

ANR8352I Requests outstanding:
ANR8326I 001: Mount 8MM volume EXP001 R/W
in drive 8MM.1 (/dev/mt0) of library
MANUALLIB within 60 minute(s).

```

Output for an Automated Library

```

ANR8352I Requests outstanding:
ANR8306I 001: Insert 3570 volume 133540 R/W into the slot with element
number 31 of library 3570LIB within 60 minutes; issue 'REPLY'
along with the request ID when ready.

```

Related commands

Table 244. Related commands for QUERY REQUEST

| Command        | Description                              |
|----------------|--|
| CANCEL REQUEST | Cancels pending volume mount requests.   |
| REPLY          | Allows a request to continue processing. |

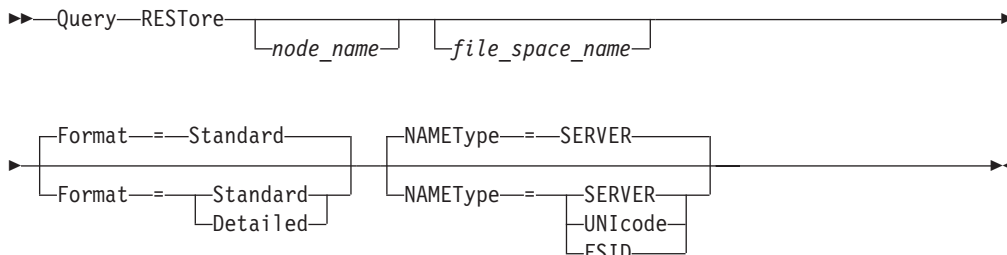
## QUERY RESTORE (Query restartable restore sessions)

Use this command to display information about the restartable restore sessions.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies the client node to be queried. This parameter is optional. If you do not specify a value, all client nodes with restartable restore sessions are displayed. You must specify a value for this parameter if you specify a file space name.

#### *file\_space\_name*

Specifies the file space to be queried. This parameter is optional. If you do not specify a value, all file spaces are matched for the specified node.

For a server that has clients with support for Unicode, you may need to have the server convert the file space name that you enter. For example, you may need to have the server convert the name you enter from the server's code page to Unicode. See the NAMETYPE parameter for details.

#### Format

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.

##### **Detailed**

Specifies that complete information is displayed.

#### NAMEType

Specify how you want the server to interpret the file space names that you enter. This parameter is useful when the server has clients with support for Unicode. You can use this parameter for Unicode-enabled Tivoli Storage Manager clients using Windows, Macintosh OS 9, Macintosh OS X, and NetWare operating systems.

Use this parameter only when you enter a partly or fully qualified file space name. The default value is SERVER. Possible values are:

##### **SERVER**

The server uses the server's code page to interpret the file space names.



### UNICODE

The server converts the file space name entered from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the name and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

### FSID

The server interprets the file space names as their file space IDs (FSIDs).

## Example: Display a restartable restore session on a specific client node

Display detailed information about client node JAMES associated with file space DRIVE\_F\_R. See "Field descriptions" for field descriptions.

```
query restore james drive_f_r format=detailed
```

```
Sess Number: -1
Restore State: Restartable
Elapsed Minutes: 2
Node Name: JAMES
FSID: 1
Filespace Name: DRIVE_F_R:
File Spec: /RESTORE/TESTDIRF\
```

## Field descriptions

### Sess Number

Specifies the session number for the restartable restore session. The number for active restore sessions is the same number displayed on the QUERY SESSION command. For restore sessions in the restartable state, a negative number is displayed for the session number. Any session number displayed in the QUERY RESTORE output may be specified from the QUERY RESTORE output.

### Restore State

- Active: Specifies the restore session is actively restoring files to the client.
- Restartable: Specifies the restore session failed and can be restarted from where it left off.

### Elapsed Minutes

Specifies the number of minutes since the restore session started. Any restartable restore session with an elapsed time greater than the RESTOREINTERVAL server option can be automatically deleted from the database when needed or during expiration processing. If the elapsed time is less than the RESTOREINTERVAL, you can delete this entry (and unlock the filesystem) only by issuing the CANCEL RESTORE command lowering the RESTOREINTERVAL value.

### Node Name

Specifies the node associated with the restartable restore session.

**FSID** Specifies the file space ID of the file space.

### Filespace Name

Specifies the file space associated with the restartable restore session.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration Center or the administrative command-line interface. The data itself is

## QUERY RESTORE

backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you may see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

### File Spec

Specifies the file specification used on the restore operation. The same file specification must be specified if a failed restore operation is to be restarted from where it stopped.

## Related commands

*Table 245. Commands related to QUERY RESTORE*

| Command        | Description                            |
|----------------|--|
| CANCEL RESTORE | Cancels a restartable restore session. |

## QUERY RPFCONTENT (Query recovery plan file contents stored on a target server)

Use this command to display the contents of a recovery plan file stored on a target server (that is, when the DEVCLASS parameter was specified on the PREPARE command). You can issue this command from either the server that created the file (the source server) or the server that stores the recovery plan file (the target server). You cannot issue this command from the server console.

The output may be delayed if the file is on tape.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>>>Query RPFContent—plan_file_name—DEVclass—=—device_class_name
                                     |
                                     |—NODENAME—=—node_name—
  
```

### Parameters

#### *plan\_file\_name* (Required)

Specifies the name of the recovery plan file to be queried. The format of the file name is *servername.yyyymmdd.hhmmss*. To see the names of existing files, issue the QUERY RPFFILE command.

#### DEVclass

Specifies the name of the device class used to create the recovery plan file. Wildcard characters are not allowed.

Specify this parameter when:

- You want to display the contents of the recovery plan file that was created for this server.
- You are issuing this command to the same server on which the PREPARE command was issued (the source server).
- The specified device class name was used on the PREPARE command that created the recovery plan file.

#### NODENAME

Specifies the node name, registered on the target server, of the source server that created the recovery plan file. Wildcard characters are not allowed.

Specify this parameter when:

- You want to display the contents of the recovery plan file that was stored on this server.
- You are issuing this command to the server that was the target of the PREPARE command that created the recovery plan file.
- The specified node name is registered to this server with a node type of SERVER.
- The Tivoli Storage Manager server that created the recovery plan file is not available.

### Example: Display the source server recovery plan

On the source server, display the contents of a recovery plan file that was created for this server on March 19, 1998, at 6:10 A.M. The PREPARE command specified the device class REMOTE. The output of this command is the entire contents of the recovery plan file. See the *Administrator's Guide* for an example of a disaster recovery plan file.

```
query rpfcontent branch1.19980319.061000 devclass=remote
```

### Example: Display the target server recovery plan

On the target server, display the contents of a recovery plan file that was stored in this server on March 19, 1998, at 6:10 A.M. The server that created the file is registered on the target server as a node named POLARIS with a node type of SERVER. The output of this command is the entire contents of the recovery plan file. See the *Administrator's Guide* for an example of a disaster recovery plan file.

```
query rpfcontent branch1.19980319.061000 nodename=polaris
```

### Related commands

Table 246. Commands related to QUERY RPFCONTENT

| Command          | Description   |
|------------------|---|
| PREPARE          | Creates a recovery plan file.   |
| QUERY RPFFILE    | Displays information about recovery plan files.                                       |
| QUERY VOLHISTORY | Displays sequential volume history information that has been collected by the server. |

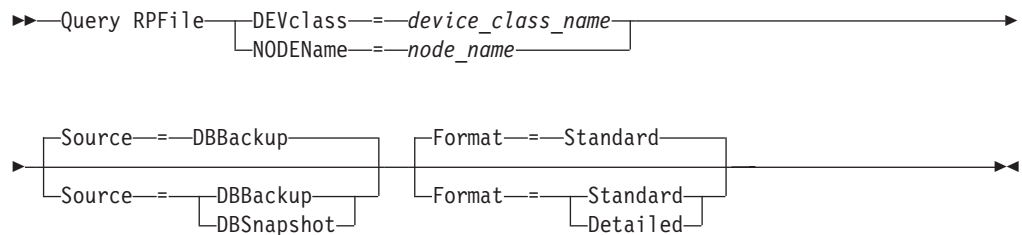
# **QUERY RPFIL (Query recovery plan file information stored on a target server)**

Use this command to display information about recovery plan files stored on a target server. You can issue this command from either the server that created the file (the source server) or the server that stores the recovery plan file (the target server).

## **Privilege class**

Any administrator can issue this command.

## **Syntax**



## **Parameters**

### **DEVclass**

Specifies the name of the device class that was used to create the recovery plan files. Use this parameter when logged on to the server that created the recovery plan file. You can use wildcard characters in the device class name. All recovery plan files that are created with the device class specified are included in the query.

### **NODENAME**

Specifies the node name, registered on the target server, of the source server that created the recovery plan files. Use this parameter when logged on to the target server. You can use this parameter when the source server is not available. You can use wildcard characters to specify the node name. All file objects that are stored with the node name specified are included in this query.

### **Source**

Specifies the type of database backup that was specified when the recovery plan file was prepared. This parameter is optional. The default is DBBACKUP. Possible values are:

#### **DBBackup**

The recovery plan file was prepared with full and incremental database backups specified.

#### **DBSnapshot**

The recovery plan file was prepared with snapshot database backups specified.

### **Format**

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

#### **Standard**

Tivoli Storage Manager displays partial information for the recovery plan file.

### Detailed

Tivoli Storage Manager displays all information for the recovery plan file.

### Example: Display detailed information about the recovery plans

Display recovery plan files that were created for this server using the specified device class. See “Field descriptions” for field descriptions.

```
query rpfile devclass=* format=detailed
```

```
Recovery Plan File Name: ALASKA.20000406.170423
Node Name: BRANCH1
Device Class Name: REMOTE
Recovery Plan File Type: RPFIL
Mgmt Class Name: STANDARD
Recovery Plan File Size: 16,255 Bytes
Marked for Deletion: Yes
Deletion Date: 06/12/2000 13:05:31

Recovery Plan File Name: ALASKA.20000407.170845
Node Name: BRANCH1
Device Class Name: REMOTE
Recovery Plan File Type: RPFSSNAPSHOT
Mgmt Class Name: STANDARD
Recovery Plan File Size: 16,425 Bytes
Marked for Deletion: No
Deletion Date:
```

### Example: Display a list of recovery plans for a specific node name

Display a list of all recovery plan file objects that are stored with the specified node name (TYPE=SERVER). See “Field descriptions” for field descriptions.

```
query rpfile nodename=branch1
```

| Recovery Plan File Name | Node Name | Device Class Name |
|-------------------------|-----------|-------------------|
| -----                   | -----     | -----             |
| ALASKA.19980406.170423  | BRANCH1   | REMOTE            |
| ALASKA.19980407.170845  | BRANCH1   | REMOTE            |

## Field descriptions

### Recovery Plan File Name

The recovery plan file name.

### Node Name

The node name that is registered with the target server and used to store the recovery plan file objects.

### Device Class Name

The device class name that is defined in the source server and used to create the recovery plan files.

### Recovery Plan File Type

The type of recovery plan file:

#### RPFIL

The plan assumes full plus incremental database backups.

#### RPFSSNAPSHOT

The plan assumes snapshot database backups.

**Mgmt Class Name**

The management class name that the recovery plan file is associated with in the target server.

**Recovery Plan File Size**

Estimated size of the recovery plan file object on the target server.

**Marked For Deletion**

Whether the object that contains the recovery plan file has been deleted from the source server and marked for deletion on the target server if the grace period has not expired. Possible values are:

**Yes**      The object is marked for deletion.

**No**        The object is not marked for deletion.

**Deletion Date**

Date that the object has been deleted from the source server and marked for deletion on the target server. This field is blank if the object has not been marked for deletion.

**Related commands**

*Table 247. Commands related to QUERY RPFIL*

| Command          | Description   |
|------------------|---|
| PREPARE          | Creates a recovery plan file.   |
| QUERY VOLHISTORY | Displays sequential volume history information that has been collected by the server. |
| QUERY RPFCONTENT | Displays the contents of a recovery plan file.  |

### QUERY SAN (Query the devices on the SAN)

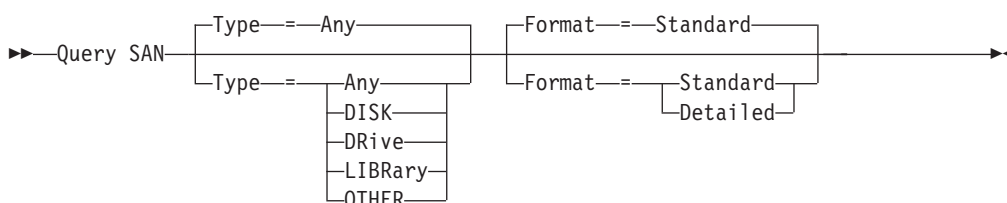
Use this command to obtain information about devices that can be detected on a storage area network (SAN), so that you can configure Tivoli Storage Manager for LAN-free data movement.

The QUERY SAN command requires the libhbaapi.a that supports SNIA common Host Bus Adapter (HBA) API. With this library object, Tivoli Storage Manager can call the hbaapi functions specified in the SNIA common HBAAPI standard.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### Type

Specifies the type of device that will be displayed. This parameter is optional. The default value is Any. Possible values are:

##### Any

Specifies that any device detected on the SAN is displayed.

##### DISK

Specifies that only disk devices are displayed.

##### DRive

Specifies that only drive devices are displayed.

##### LIBRARY

Specifies that only library devices are displayed.

##### OTHER

Specifies that devices that are not disks, drives, or libraries are displayed.

##### Format

Specifies the type of information that will be displayed. This parameter is optional. The default value is Standard. Possible values are:

##### Standard

Specifies that the information displayed will be summarized.

##### Detailed

Specifies that complete information will be displayed.

**Tip:** The output might not display the serial number of the device. If this happens, look on the back of the device or contact the manufacturer of the device.



### Example: List drive devices

Display summary information for drive devices on a SAN. See “Field descriptions” for field descriptions.

```
query san type=drive
```

| Device Type | Vendor  | Product | Serial       | Device    |
|-------------|---------|---------|--------------|-----------|
| -----       | -----   | -----   | -----        | -----     |
| LIBRARY     | STK     | L180    | MPC01000128  | /dev/smc1 |
| DRIVE       | STK     | 9840    | 331001017229 | /dev/rmt3 |
| DRIVE       | Quantum | DLT4000 | JF62806275   | /dev/rmt4 |
| DRIVE       | Quantum | DLT4000 | JP73213185   | /dev/rmt5 |
| DRIVE       | STK     | 9840    | 331000028779 | /dev/rmt6 |

### Example: Display drive device information

Display detailed information for all drive devices on a SAN. See “Field descriptions” for field descriptions.

```
query san type=drive format=detailed
```

```

Device Type:  DRIVE
Vendor:       IBM
Product:      03570B02
Serial:
Device:       mt10.2.0.3
DataMover:    No
Node WWN:     5005076206039E05
Port WWN:     5005076206439E05
Lun:          0
SCSI Port:    3
SCSI Bus:     0
SCSI Target:  10

```

### Field descriptions

#### Device Type

The type of device that is being displayed.

#### Vendor

The name of the device’s vendor.

#### Product

The name of the product assigned by the vendor.

**Serial** The serial number of the device.

#### Device

The device special file name.

#### Data Mover

Whether the device is a data mover.

#### Node WWN

The worldwide name for the device.

#### Port WWN

The worldwide name for the device, which is specific to the port that the device is connected to.

**Lun** The Logical Unit Number of the device.

#### SCSI Port

The port of the Fibre Channel (or SCSI) Host Bus Adapter.

## QUERY SAN

### SCSI Bus

The bus of the Host Bus Adapter card.

### SCSI Target

The target number of the device.

## Related commands

*Table 248. Commands related to QUERY SAN*

| Command          | Description  |
|------------------|--|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server. |
| DEFINE DRIVE     | Assigns a drive to a library.                                  |
| DEFINE LIBRARY   | Defines an automated or manual library.                        |

## QUERY SCHEDULE (Query schedules)

Use this command to display information about 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 operation are defined separately. Some options in the query display will be blank depending on whether the schedule style is classic or enhanced.

*Table 249. Commands related to QUERY SCHEDULE*

| Command         | Description   |
|-----------------|---|
| COPY SCHEDULE   | Creates a copy of a schedule.   |
| DEFINE SCHEDULE | Defines a schedule for a client operation or an administrative command. |
| UPDATE SCHEDULE | Changes the attributes of a schedule.                                   |

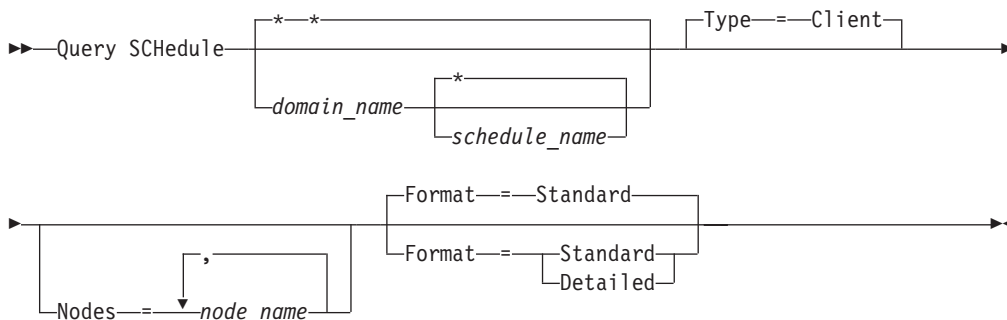
### QUERY SCHEDULE (Query client schedules)

Use this command to display information about one or more client schedules.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *domain\_name*

Specifies the name of the policy domain to which the schedule belongs. You can use a wildcard character to specify this name. If you specify a domain name, you do not have to specify a schedule name.

##### *schedule\_name*

Specifies the name of the schedule that belongs to the specified policy domain. You can use a wildcard character to specify this name. If you specify a schedule name, you must also specify a policy domain name.

##### **Type=Client**

Specifies that the query displays client schedules. This parameter is optional. The default is CLIENT.

##### **Nodes**

Specifies the name of one or more client nodes that are associated with the schedules to be displayed. This parameter is optional. You can use a wildcard character to specify client nodes. If you do not specify a client name, all schedules matching the DOMAINNAME and SCHEDULENAME parameters are displayed. You can specify multiple client nodes by separating the names with commas and no intervening spaces.

##### **Format**

Specifies how information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the schedules.

##### **Detailed**

Specifies that detailed information is displayed for the schedules.

The standard format displays a blank in the period column and an asterisk in the day column for enhanced schedules. To display complete information about an enhanced schedule, issue FORMAT=DETAILED.

**Example: List schedules for a specific policy domain**

Display all schedules that belong to the EMPLOYEE\_RECORDS policy domain. See “Field descriptions: Schedules for a specific policy domain” for field descriptions.

```
query schedule employee_records
```

The standard format displays a blank in the period column and an asterisk in the day column for enhanced schedules. To display complete information about an enhanced schedule, issue `FORMAT=DETAILED`.

| Domain                     | * | Schedule Name   | Action | Start Date/Time     | Duration | Period | Day |
|----------------------------|---|-----------------|--------|---------------------|----------|--------|-----|
| EMPLOY-<br>EE_RE-<br>CORDS |   | WEEKLY_BACKUP   | Inc Bk | 2004.06.04 17.04.20 | 1 H      | 1 D    | Any |
| EMPLOY-<br>EE_RE-<br>CORDS |   | EMPLOYEE_BACKUP | Inc Bk | 2004.06.04 17.04.20 | 1 H      |        | (*) |

**Field descriptions: Schedules for a specific policy domain**

**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 is processed.

**Start Date/Time**

Specifies the initial starting date and time for this schedule.

**Duration**

Specifies the length of the startup window for this schedule.

**Period** Specifies the time between startup windows (assuming DAYOFWEEK=ANY). The column is blank for enhanced schedules.

**Day** Specifies the day of the week on which the startup windows for the schedule begin. The column contains an asterisk for enhanced schedules.

**Example: Display detailed client schedules**

From a managed server, display detailed information about client schedules. See “Field descriptions: Detailed client schedules” on page 760 for field descriptions.

```
query schedule * type=client format=detailed
```

## QUERY SCHEDULE

```
Policy Domain Name: ADMIN_RECORDS
Schedule Name: ADMIN_BACKUP
Description:
  Action: Incremental
  Options:
  Objects:
  Priority: 5
Start Date/Time: 2004.06.04 17.04.20
Duration: 1 Hour(s)
Schedule Style: Classic
Period: 1 Day(s)
Day of Week: Any
Month:
Day of Month:
Week of Month:
Expiration:
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 2004.06.04 17.51.49
Managing profile: ADMIN_INFO
Policy Domain Name: EMPLOYEE_RECORDS
Schedule Name: EMPLOYEE_BACKUP
Description:
  Action: Incremental
  Options:
  Objects:
  Priority: 5
Start Date/Time: 2004.06.04 17.04.33
Duration: 1 Hour(s)
Schedule Style: Enhanced
Period:
Day of Week: Any
Month: Mar,Jun,Nov
Day of Month: -14,14,22
Week of Month: Last
Expiration:
Last Update by (administrator): $$CONFIG_MANAGER$$
Last Update Date/Time: 2004.06.04 17.18.30
Managing profile: EMPLOYEE
```

### Field descriptions: Detailed client schedules

#### Policy Domain Name

Specifies the name of the policy domain.

#### Schedule Name

Specifies the name of the schedule.

#### Description

Specifies the description of the schedule.

#### Action

Specifies the type of action that occurs when this schedule is run. See the DEFINE SCHEDULE command for a listing of actions.

#### Options

Specifies the options that are supplied to the DSMC command when the schedule is run.

#### Objects

Specifies the objects for which the specified action is performed.

#### Priority

Specifies the priority value for the schedule.

#### Start Date/Time

Specifies the initial starting date and time for the schedule.

### **Duration**

Specifies the length of the startup window for the schedule.

### **Schedule Style**

Specifies whether classic or enhanced schedule rules are used.

### **Period**

Specifies the time between startup windows (assuming DAYOFWEEK=ANY). This is not displayed for enhanced syntax schedules.

### **Day of Week**

Specifies the day of the week on which the startup windows for the schedule begin. Using a standard format displays an asterisk in the day of week field for enhanced schedules.

### **Month**

Specifies the months during which the schedule will run. This is not displayed for classic syntax schedules.

### **Day of Month**

Specifies the days of the month during which the schedule will run. This is not displayed for classic syntax schedules.

### **Week of Month**

Specifies the weeks (first, second, third, fourth, or last) of the month during which the schedule will run. This is not displayed for classic syntax schedules.

### **Expiration**

Specifies the date and time on which this schedule expires. If this column is blank, the schedule does not expire.

### **Last Update by (administrator)**

Specifies the name of the administrator that most recently updated the schedule. If this field contains a \$\$CONFIG\_MANAGER\$\$, the schedule is associated with a domain that is managed by the configuration manager.

### **Last Update Date/Time**

Specifies the last date and time the schedule was last updated.

### **Managing Profile**

Specifies the profile or profiles to which the managed server subscribed to get the definition of this schedule.

## QUERY SCHEDULE

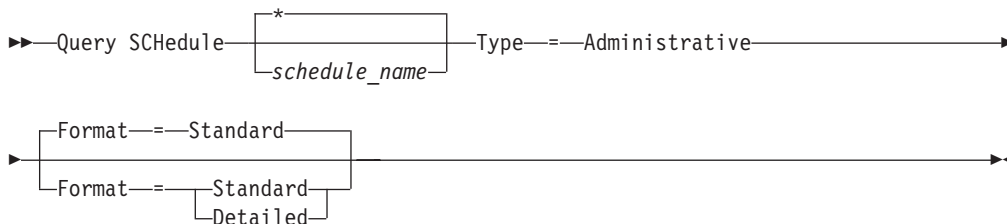
### QUERY SCHEDULE (Query an administrative schedule)

Use this command to display information about one or more administrative schedules.

#### Privilege class

Any administrator can issue this command.

#### Syntax



#### Parameters

##### *schedule\_name*

Specifies the name of the schedule to be queried. You can use a wildcard character to specify this name.

##### **Type=Administrative (Required)**

Specifies that the query displays administrative command schedules.

##### **Format**

Specifies how information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information is displayed for the schedules.

##### **Detailed**

Specifies that detailed information is displayed for the schedules.

The standard format displays a blank period column and an asterisk in the day column for enhanced schedules. Issue `FORMAT=DETAILED` to display complete information about an enhanced schedule.

### Example: Display detailed information on administrative command schedules

From a managed server, display detailed information about administrative command schedules. See “Field descriptions” on page 763 for field descriptions.

```
query schedule * type=administrative  
format=detailed
```



```

Schedule Name: BACKUP_ARCHIVEPOOL
Description:
  Command: backup db
  Priority: 5
Start Date/Time: 2004.06.04 16.57.15
Duration: 1 Hour(s)
Schedule Style: Classic
Period: 1 Day(s)
Day of Week: Any
Month:
Day of Month:
Week of Month:
Expiration:
  Active: No
Last Update by (administrator): $$CONFIG MANAGER$$
Last Update Date/Time: 2004.06.04 17.51.49
Managing Profile: ADMIN_INFO
Schedule Name: MONTHLY_BACKUP
Description:
  Command: q status
  Priority: 5
Start Date/Time: 2004.06.04 16.57.14
Duration: 1 Hour(s)
Schedule Style: Enhanced
Period:
Day of Week: Tue,Thu,Fri
Month: Aug,Nov
Day of Month:
Week of Month: Second,Third
Expiration:
  Active: No
Last Update by (administrator): $$CONFIG MANAGER
Last Update Date/Time: 2004.06.04 17.51.49
Managing Profile: ADMIN_INFO

```

## Field descriptions

### Schedule Name

Specifies the name of the schedule.

### Description

Specifies the description of the schedule.

### 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.

### Schedule Style

Specifies whether classic or enhanced schedule rules are used.

**Period** Specifies the time between startup windows (assuming DAYOFWEEK=ANY). This is not displayed for enhanced syntax schedules.

### Day of Week

Specifies the day of the week on which the startup windows begin.

### Month

Specifies the months during which the schedule will run. This is not displayed for classic syntax schedules.

## QUERY SCHEDULE

### **Day of Month**

Specifies the days of the month during which the schedule will run. This is not displayed for classic syntax schedules.

### **Week of Month**

Specifies the weeks (first, second, third, fourth, or last) of the month during which the schedule will run. This is not displayed for classic syntax schedules.

### **Expiration**

Specifies the date after which this schedule will no longer be used. If this column is blank, the schedule does not expire.

**Active** Specifies whether the schedule has been processed according to the time and date set for this schedule.

### **Last Update by (administrator)**

Specifies the name of the administrator that most recently updated the schedule. If this field contains a \$\$CONFIG\_MANAGER\$\$, the schedule is associated with a domain that is managed by the configuration manager.

### **Last Update Date/Time**

Specifies the last date and time the schedule was modified.

### **Managing Profile**

Specifies the profile or profiles to which the managed server subscribed to get the definition of this schedule.

## QUERY SCRIPT (Query Tivoli Storage Manager scripts)

Use this command to display information about scripts.

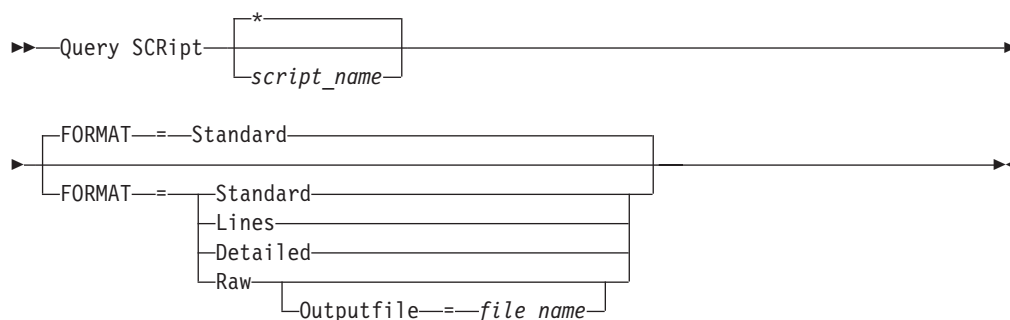
You can use this command in conjunction with the DEFINE SCRIPT command to create a new script using the contents from another script.

### Privilege class

The privilege class required for this command depends on whether the OUTPUTFILE parameter is specified in the command.

- If the OUTPUTFILE parameter is not specified, any administrator can issue this command.
- If the OUTPUTFILE parameter is specified and the REQSYSAUTHOUTFILE server option is set to YES, the administrator must have system privilege.
- If the OUTPUTFILE parameter is specified and the REQSYSAUTHOUTFILE server option is set to NO, the administrator must have operator, policy, storage, or system privilege.

### Syntax



### Parameters

#### *script\_name*

Specifies the name of the script for which information is to be displayed. You can include a wildcard character to specify this name.

**Important:** If you do not specify a script, the query displays information about all scripts. The time used to process this command and the amount of information displayed can be extensive.

#### Format

Specifies the output format for displaying script information. The default is STANDARD. Possible values are:

##### Standard

Specifies that only the script name and description in a script are displayed.

##### Lines

Specifies that the script name, the line number of the commands, comment lines, and the commands in the script are displayed.

##### Detailed

Specifies that detailed information about the script is displayed.

## QUERY SCRIPT

### Raw

Specifies that commands contained in the script are written to a file named with the OUTPUTFILE parameter. This format is a way of directing output from a script to a file so that it can be copied into another script using the DEFINE SCRIPT command.

If no output file is specified, the Tivoli Storage Manager server outputs the "query script" with "format=raw" to the console.

### Outputfile

Specifies the name of the file to which output is directed when you specify FORMAT=RAW. The file you specify must reside on the server running this command. If the file exists, the query output is appended to the end of the file.

### Example: List the script descriptions

Display the standard information about scripts.

```
query script *
```

| Name    | Description  |
|---------|--|
| -----   | -----  |
| QCOLS   | Display columns for a specified SQL table            |
| QSAMPLE | Sample SQL Query                                     |
| EXAMPLE | Backup the store pools and database when no sessions |

### Example: Display the contents of a script with line numbers

Display the lines information for a script named Q\_AUTHORITY.

```
query script q_authority format=lines
```

| Name        | Line Number | Command   |
|-------------|-------------|---|
| -----       | -----       | -----   |
| Q_AUTHORITY | 1           | /* -----*/                                      |
|             | 5           | /* Script Name: Q_AUTHORITY */                  |
|             | 10          | /* Description: Display administrators that */  |
|             | 15          | /* have the authority to issue */               |
|             | 20          | /* commands requiring a */                      |
|             | 25          | /* specific privilege. */                       |
|             | 30          | /* Parameter 1: privilege name - in the form */ |
|             | 35          | /* x_priv - EX. policy_priv */                  |
|             | 40          | /* Example: run q_authority storage_priv */     |
|             | 45          | /* -----*/                                      |
|             | 50          | select admin_name from admins where -           |
|             | 55          | upper(system_priv) <> 'NO' or -                 |
|             | 60          | upper(\$1) <> 'NO'                              |

### Example: Create a script from an existing script

Query the ENGDEV script and direct the output to a file named MY.SCRIPT.

```
query script engdev format=raw outputfile=my.script
```

### Example: Create a script from an existing script

Create a new script by using the contents from file, MY.SCRIPT. Name the new script AGADM.

```
define script agadm file=my.script
```

## Example: Display detailed script information

Display detailed information about scripts. See “Field descriptions” for field descriptions.

```
query script * format=detailed
```

```

      Name: QCOLS
      Line Number: DESCRIPTION
      Command: Display columns for a specified SQL
                table
Last Update by (administrator): SERVER_CONSOLE
      Last Update Date/Time: 12/02/1997 16:05:29

      Name: QCOLS
      Line Number: 1
      Command: select colname from columns where
                tabname='$1'
Last Update by (administrator): SERVER_CONSOLE
      Last Update Date/Time: 12/02/1997 16:05:29

```

## Field descriptions

**Name** The name of the script.

**Line Number**

The line number of the script or the string DESCRIPTION.

**Command**

The command included on the line number displayed in the previous field.

**Last Update by (administrator)**

The name of the administrator that has defined or most recently updated the script.

**Last Update Date/Time**

The date and time that the administrator defined or updated the script.

## Related commands

Table 250. Commands related to QUERY SCRIPT

| Command       | Description  |
|---------------|--|
| COPY SCRIPT   | Creates a copy of a script.                                |
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| DELETE SCRIPT | Deletes the script or individual lines from the script.    |
| RENAME SCRIPT | Renames a script to a new name.                            |
| RUN           | Runs a script.   |
| UPDATE SCRIPT | Changes or adds lines to a script.                         |

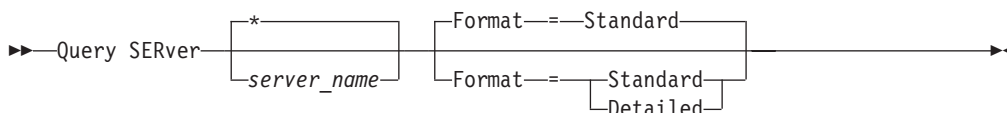
## QUERY SERVER (Query a server)

Use this command to display information about a server definition.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *server\_name*

Specifies the name of the server to be queried. You can use wildcard characters to specify this name. This parameter is optional. The default is all server names.

#### Format

Specifies how the information is displayed. The parameter is optional. The default is STANDARD.

#### Standard

Specifies that partial information is displayed.

#### Detailed

Specifies that complete information is displayed.

### Example: List all servers

Display information in standard format about all servers. See “Field descriptions” on page 769 for field descriptions.

query server \*

| Server Name | Comm. Method | High-level Address | Low-level Address | Days Since Last Access | Server Password Set | Virtual Volume Password Set | Allow Replacement |
|-------------|--------------|--------------------|-------------------|------------------------|---------------------|-----------------------------|-------------------|
| SERVER_A    | TCPIP        | 9.115.35.6         | 1501              | 11                     | Yes                 | No                          | No                |
| SERVER_B    | TCPIP        | 9.115.45.24        | 1500              | <1                     | Yes                 | No                          | No                |
| ASTRO       | TCPIP        | 9.115.32.21        | 1500              | 24                     | Yes                 | No                          | No                |

### Example: Display detailed information on a specific server

From a managed server, display detailed information about SERVER\_A. See “Field descriptions” on page 769 for field descriptions.

query server server\_a format=detailed

```

Server Name: SERVER_A
Comm. Method: TCPIP
High-level Address: 9.115.4.15
Low-level Address: 1500
Description:
Allow Replacement:
Node Name:
Last Access Date/Time: 06/09/1998 12:51:51
Days Since Last Access: <1
Locked?:
Compression: Client's Choice
Archive Delete Allowed?: No
URL:
Registration Date/Time: 06/09/1998 12:51:51
Registering Administrator: $$CONFIG_MANAGER$$
Bytes Received Last Session: 0
Bytes Sent Last Session: 0
Duration of Last Session (sec): 0.00
Pct. Idle Wait Last Session: 0.00
Pct. Comm. Wait Last Session: 0.00
Pct. Media Wait Last Session: 0.00
Grace Deletion Period: 5
Managing Profile:
Server Password Set Date/Time:
Days since Server Password Set:
Invalid Sign-on count for Server:
Virtual Volume Password Set: No
Virtual Volume Password Set Date/Time:(?)
Days Since Virtual Volume Password Set:(?)
Invalid Sign-on Count for Virtual Volume Node: 0
Validate Protocol: No

```

## Field descriptions

### Server Name

The name of the server.

### Communication Method

The communication method used to connect to the server.

### High-level Address

The IP address (in dotted decimal format) of the server.

### Low-level Address

The port number of the server.

### Description

The server description.

### Allow Replacement

Whether a server definition on a managed server can be replaced with a definition from a configuration manager.

### Node Name

The name of the client node.

### Last Access Date/Time

The last date and time that the client node accessed the server.

### Days Since Last Access

The number of days since the client node accessed the server.

### Locked?

Whether the client node is locked out of Tivoli Storage Manager.

**Compression**

The type of compression performed by Tivoli Storage Manager on client files.

**Archive Delete Allowed?**

Whether the client node can delete its own archive files. A value of (?) denotes that this field is not set and does not apply to this definition.

**URL** The URL used to access this server from a Web browser-based interface.

**Registration Date/Time**

The date and time that the client node was registered.

**Registering Administrator**

The name of the administrator that registered the client node.

**Bytes Received Last Session**

the number of bytes received by the server during the last client node session.

**Bytes Sent Last Session**

The number of bytes sent to the client node.

**Duration of Last Session (sec)**

The length of the last client node session, in seconds.

**Pct. Idle Wait Last Session**

The percentage of the total session time that the client was not performing any functions.

**Pct. Comm. Wait Last Session**

The percentage of the total session time that the client waited for a response from the server.

**Pct. Media Wait Last Session**

The percentage of the total session time that the client waited for a removable volume to be mounted.

**Grace Deletion Period**

How many days an object remains on the target server after it has been marked for deletion.

**Managing Profile?**

The profile from which the managed server got the definition of this server.

**Server Password Set Date/Time**

When the server's password was set.

**Days since Server Password Set**

The number of days since the server password was set.

**Invalid Sign-on count for Server**

The maximum number of invalid sign-on attempts that the server will accept.

**Virtual Volume Password Set**

Whether the password used to sign on to the target server has been set.

**Virtual Volume Password Set Date/Time**

When the password for virtual volume support was last set.

**Days Since Virtual Volume Password Set**

The number of days since the password for virtual volume support was set.



### Invalid Sign-on Count for Virtual Volume Node

The maximum number of invalid sign-on attempts that are accepted on the target server.

### Validate Protocol

Specifies whether the storage agent has data validation enabled.

## Related commands

Table 251. Commands related to QUERY SERVER

| Command           | Description  |
|-------------------|--|
| DEFINE DEVCLASS   | Defines a device class.  |
| DEFINE SERVER     | Defines a server for server-to-server communications.                                  |
| DELETE DEVCLASS   | Deletes a device class name.   |
| DELETE FILESPACE  | Deletes data associated with client's file spaces.                                     |
| DELETE SERVER     | Deletes the definition of a server.  |
| QUERY NODE        | Displays partial or complete information about one or more clients.                    |
| RECONCILE VOLUMES | Reconciles source server virtual volume definitions and target server archive objects. |
| REGISTER NODE     | Defines a client to the server and sets options for that user.                         |
| REMOVE NODE       | Removes a client from the list of registered nodes for a specific policy domain.       |
| UPDATE DEVCLASS   | Changes the attributes of a device class.  |
| UPDATE NODE       | Changes the attributes associated with a client node.                                  |
| UPDATE SERVER     | Updates information about a server.  |

## QUERY SERVERGROUP (Query a server group)

Use this command to display information about server groups and group members.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

*group\_name*

Specifies the server group to query. This parameter is optional. You can use wildcard characters to specify this name.

### Example: List server groups

From a managed server, query all server groups. “Field descriptions” for field descriptions.

```
query servergroup *
```

| Server Group | Group Member                                 | Description  | Managing Profile |
|--------------|--|--------------|------------------|
| ADMIN_GROUP  | SERVER_A<br>SERVER_B<br>SERVER_C<br>SERVER_D | Headquarters | ADMIN_INFO       |

### Field descriptions

#### Server Group

The name of the server group.

#### Group Member

The group members.

#### Description

The description of the server group.

#### Managing Profile

The profile or profiles to which the managed server subscribed to get the definition of the server groups.

### Related commands

Table 252. Commands related to QUERY SERVERGROUP

| Command            | Description                         |
|--------------------|-------------------------------------|
| COPY SERVERGROUP   | Creates a copy of a server group.   |
| DEFINE SERVERGROUP | Defines a new server group.         |
| DELETE SERVERGROUP | Deletes a server group.             |
| QUERY SERVER       | Displays information about servers. |

*Table 252. Commands related to QUERY SERVERGROUP (continued)*

| Command            | Description             |
|--------------------|-------------------------|
| RENAME SERVERGROUP | Renames a server group. |
| UPDATE SERVERGROUP | Updates a server group. |

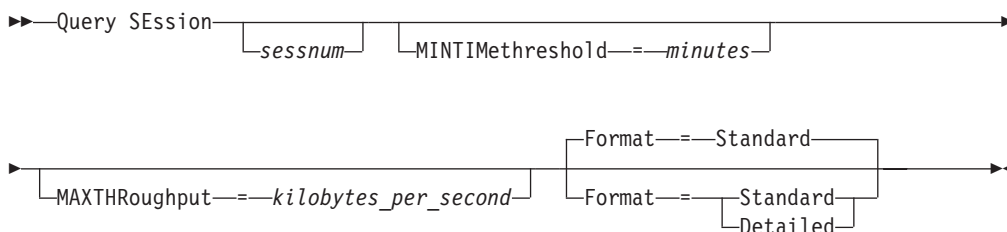
### QUERY SESSION (Query client sessions)

Use this command to display information about 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 query. This parameter is optional. If you do not specify a value for this parameter, all sessions display.

##### **MINTIMethreshold**

Specifies to display sessions that have had at least this number of minutes elapse from the time the client sent data to the server for storage. This parameter is optional. The minimum number of minutes is 1. The maximum number of minutes is 99999999.

##### **MAXTHRoughput**

Specifies to display sessions that are transferring data at a rate less than this number of kilobytes per second. This parameter is optional. The minimum number of kilobytes per second is 0. The maximum number of kilobytes per second is 99999999.

##### **Format**

Specifies how the information displays. This parameter is optional. The default value is STANDARD. Possible values are:

##### **Standard**

Specifies that partial information displays for the session.

##### **Detailed**

Specifies that complete information displays for the session.

#### Example: List active client node sessions

Display information on all administrative and client node sessions that are communicating with the server. See “Field descriptions” on page 775 for field descriptions.

```
query session
```

| Sess<br>Number | Comm.<br>Method | Sess<br>State | Wait<br>Time | Bytes<br>Sent | Bytes<br>Recvd | Sess<br>Type | Platform | Client<br>Name |
|----------------|-----------------|---------------|--------------|---------------|----------------|--------------|----------|----------------|
| 77             | TCP/IP          | Run           | 0 S          | 134           | 159            | Admin        | OS/2     | ADMIN          |

## Example: Display detailed information on active client node sessions

Display information in full detail about all administrative and client node sessions that are communicating with the server. See “Field descriptions” for field descriptions.

query session format=detailed

```

Sess Number: 77
Comm. Method: Tcp/Ip
Sess State: MediaW
Wait Time: 13 S
Bytes Sent: 2.2 K
Bytes Recvd: 1.8 K
Sess Type: Node
Platform: AIX
Client Name: N25XY
Media Access Status: Waiting for mount of output volume:
                     COPYVOL1, 8MMPPOOL ( 12 seconds )

                     Waiting for mount point:
                     8MMCLASS, ONSITEPOOL ( 45 seconds )

                     Waiting for output volume:
                     COPYVOL2, OFFSITEPOOL ( 30 seconds )
Proxy By Storage Agent: mickeynode
User Name:
Date/Time First Data Sent: 07/08/08      18:01:21
Actions: BKIns

```

## Field descriptions

### 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:

**End** The session is ending (session resources are released).

**IdleW** Waiting for client’s next request (session is idle).

### MediaW

The session is waiting for access to a sequential access volume.

### RecvW

Waiting to receive an expected message from the client.

### Run

The server is executing a client request (and not waiting to send data).

### SendW

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).

## QUERY SESSION

**Start** 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: ADMIN for an administrative session, NODE for a client node session, or SERVER. SERVER specifies the server starts a session and initiates server-to-server operations such as central configuration, library sharing, and storage agent sessions.

**Platform**

Specifies the type of operating system associated with the client.

**Client Name**

Specifies the name of the client node or the administrator.

**Media Access Status**

Specifies the type of media wait state. This information is only provided when the session state equals MediaW. When a session is in a media wait state, this field displays a list of all mount points and sequential volumes for the session. The list of mount points specifies the device class and the associated storage pool. The list of volumes specifies the primary storage pool volumes in addition to any copy storage pool and active-data pool volumes along with their assigned storage pool.

The server allows multiple sessions to read and one session to write to a volume concurrently in a storage pool associated with the FILE device type. As a result, a volume in a storage pool with a device type of FILE can appear as the current volume for more than one session.

**Proxy by Storage Agent**

Specifies the storage agent that is the proxy for LAN-Free data movement for the node.

**User Name**

Specifies the user ID of the node, on a multi-user system, that connects to the server when it is not the same system user who originally connected to the server.

**Date/Time First Data Sent**

Specifies the date and time that the client first sent data to the server for storage.

**Actions**

Displays a list of actions that have been performed during the session. An action is listed only once, even if the action occurs multiple times during a session. Possible actions are:

**BkIns** One or more backup objects were stored on the server. The operation might have been an incremental backup or a selective backup.

**BkUpd**

One or more attributes were updated for a backup object that is stored on the server.

**BkDel** One or more backup objects stored on the server have been deleted.

**BkRebind**

One or more backup objects stored on the server have been bound to a different management class.

**NoQueryRestore**

A no-query restore operation was initiated from the client, to restore backed-up files from the server to the client system.

**ArIns** One or more archive objects were stored on the server.

**ObjRtrv**

One or more files were retrieved from the server. This might have been to retrieve archive files, or to restore backup data (except for backup data from a no-query restore operation).

**MigIns**

One or more files have been migrated and stored on the server by Tivoli Storage Manager for Space Management (HSM client).

**MigDel**

One or more space-managed files that were stored on the server have been deleted.

**MigRebind**

One or more space-managed files that are stored on the server have been bound to a different management class.

**MigRecall**

One or more space-managed files that are stored on the server have been recalled.

**MigUpd**

The attributes for one or more space-managed files that are stored on the server have been updated.

**FSAdd**

The client node added one or more new file spaces to server storage.

**FSUpd**

The client node updated attributes for one or more file spaces that are defined to the server.

**DefAuth**

A SET ACCESS command has been processed by the client node, which caused an authorization rule for access to the client node's data to be added.

**Related commands**

*Table 253. Command related to QUERY SESSION*

| Command        | Description                              |
|----------------|--|
| CANCEL SESSION | Cancels active sessions with the server. |

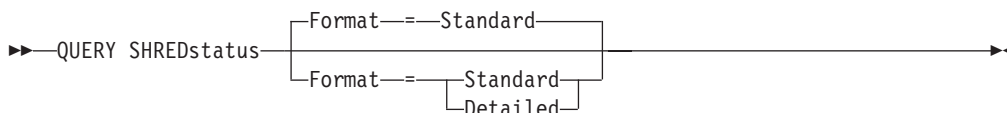
## QUERY SHREDSTATUS (Query shredding status )

Use this command to display information about data waiting to be shredded.

### Privilege class

To issue this command you must have administrator privilege.

### Syntax



### Parameters

#### Format

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

##### Standard

Specifies that partial information is displayed. This is the default.

##### Detailed

Specifies that complete information is displayed.

### Example: Display summary shredding information

Show partial information about data shredding on the server. See “Field descriptions” for field descriptions.

```
query shredstatus
```

| Shredding | Objects  |
|-----------|----------|
| Active    | Awaiting |
|           | Shred    |
| -----     | -----    |
| NO        | 4        |

### Example: Display detailed shredding information

Display detailed information about data shredding on the server. See “Field descriptions” for field descriptions.

```
query shredstatus format=detailed
```

| Shredding | Objects | Occupied | Data Left |
|-----------|---------|----------|-----------|
| Active    | waiting | Space    | To Shred  |
|           | Shred   | (MB)     | (MB)      |
| -----     | -----   | -----    | -----     |
| NO        | 4       | 182      | 364       |

### Field descriptions

#### Shredding Active

Indicates whether or not the server is actively shredding data at this time.

#### Objects Awaiting Shred

The number of objects currently waiting to be shredded.



### Occupied Space (MB)

The amount of server storage space occupied by the objects currently waiting to be shredded, in megabytes. This is the amount of space that will become available when the objects are shredded.

### Writes to Complete Shred (MB)

The amount of data that must be written in order to complete the shredding, in megabytes. This includes multiple overwrites. In the previous example on displaying detailed shredding information, the objects currently waiting to be shredded occupy 182 MB of space and the overwrite count is 2, so a total of 364 MB must be written in order to shred the objects.

## Related commands

Table 254. Commands related to QUERY SHREDSTATUS

| Command               | Description   |
|-----------------------|---|
| BACKUP STGPOOL        | Backs up a primary storage pool to a copy storage pool.                         |
| DEFINE STGPOOL        | Defines a storage pool as a named collection of server storage media.           |
| EXPORT NODE           | Copies client node information to external media.                               |
| GENERATE BACKUPSET    | Generates a backup set of a client's data.                                      |
| GENERATE BACKUPSETTOC | Generates a table of contents for a backup set.                                 |
| MOVE DATA             | Moves data from a specified storage pool volume to another storage pool volume. |
| QUERY STGPOOL         | Displays information about storage pools.                                       |
| SETOPT                | Updates a server option without stopping and restarting the server.             |
| SHRED DATA            | Manually starts the process of shredding deleted data.                          |
| UPDATE STGPOOL        | Changes the attributes of a storage pool.                                       |

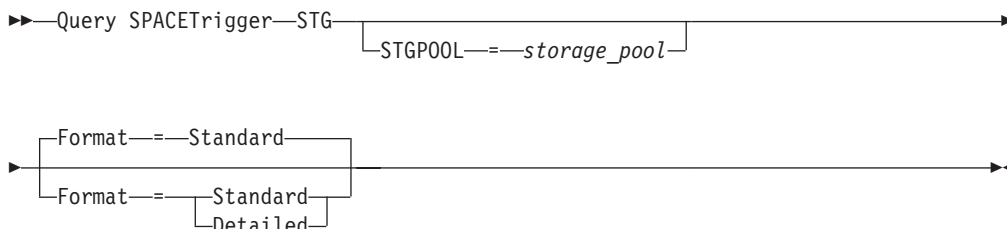
## QUERY SPACETRIGGER (Query the space triggers)

Use this command to display the settings for storage pool space triggers.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### STG

Specifies a storage pool space trigger

#### STGPOOL

Specifies one or more storage pools (using a wildcard) for which storage pool trigger information will be displayed. If STG is specified but STGPOOL is not, the default storage pool space trigger, if any, is displayed.

#### Format

Specifies how the information is displayed. This parameter is optional. The default is STANDARD. Possible values are:

##### Standard

Specifies that partial information is displayed.

##### Detailed

Specifies that complete information is displayed.

### Example: Display detailed settings for a storage pool space trigger

Issue this command:

```
query spacetrigger stg stgpool=archivepool format=detailed
```

```

STGPOOL Full Percentage: 50
STGPOOL Expansion Percentage: 20
STGPOOL Expansion prefix: /usr/tivoli/tsm/server/filevol/
STGPOOL: ARCHIVEPOOL
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 05/10/2004 11:59:59
  
```

### Field descriptions

#### STGPOOL Full Percentage

The trigger utilization percentage at which Tivoli Storage Manager allocates more space for the storage pool.

#### STGPOOL Space Expansion

The percentage of space by which the storage pool should be expanded.

**STGPOOL Expansion prefix**

The prefix associated with the space trigger.

**STGPOOL**

The storage pool name associated with the query.

**Last Update by (administrator)**

The administrator who last updated the storage pool space trigger.

**Last Update Date/Time**

The date and time when the administrator last updated the storage pool space trigger.

**Related commands**

*Table 255. Commands related to QUERY SPACETRIGGER*

| Command             | Description   |
|---------------------|---|
| DEFINE SPACETRIGGER | Defines a space trigger to expand the space for a storage pool. |
| DELETE SPACETRIGGER | Deletes the storage pool space trigger.                         |
| UPDATE SPACETRIGGER | Changes attributes of storage pool space trigger.               |

### QUERY STATUS (Query system parameters)

Use the QUERY STATUS command to display information about system parameters.

Use this command for the following:

- To display the service level of the server
- To display information about the general server parameters, such as those defined by the SET commands
- 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
- To display information about the central scheduler, such as the central scheduling mode of the server
- To display the maximum number of retries that are allowed after a failed attempt to execute a scheduled command
- To display whether subfiles can be backed up to this server, as indicated by the SET SUBFILE command.

#### Privilege class

Any administrator can issue this command.

#### Syntax

►►—Query Status—◄◄

#### Parameters

None.

#### Example: Query the status of a configuration manager

Display general information about server parameters. The command is issued from a configuration manager. See “Field descriptions” on page 783 for field descriptions.

```
query status
```

```

Server Name: SERVER1
Server host name or IP address: 9.115.44.121
Server TCP/IP port number: 5511
Server URL:
Crossdefine: Off
Server Password Set: Yes
Server Installation Date/Time: October 12, 2008 07:18:16 AM
Server Restart Date/Time: October 12, 2008 01:05:35 PM
Authentication: On
Password Expiration Period: 90 Day(s)
Invalid Sign-on Attempt Limit: 0
Minimum Password Length: 0
Registration: Closed
Subfile Backup: No
Availability: Enabled
Accounting: Off
Activity Log Retention: 1 Day(s)
Activity Summary Retention Period:
Activity Log Number of Records: 15,623
Activity Log Size: 18 M
License Audit Period: 30 Day(s)
Last License Audit: October 12, 2008 07:43:35 AM
Server License Compliance: Valid
Central Scheduler: Active
Maximum Sessions: 25
Maximum Scheduled Sessions: 12
Event Record Retention Period: 10 Day(s)
Client Action Duration: 5 Day(s)
Schedule Randomization Percentage: 25
Query Schedule Period: Client's Choice
Maximum Command Retries: Client's Choice
Retry Period: Client's Choice
Scheduling Modes: Any
Active Receivers: CONSOLE ACTLOG
Configuration manager?: On
Refresh interval: 60
Last refresh date/time: October 12, 2008 07:43:45 AM
Context Messaging: On
Table of Contents (TOC) Load Retention: 120 Minute(s)
Machine Globally Unique ID:
Archive Retention Protection: Off
Database Reporting Mode : Partial
Database Directories : /space/tribble/db2/cn_db2/dbpath1,
                        /space/tribble/db2/cn_db2/dbpath2
Database Total Space : 3,497,600
Database Used Space : 2,856,645.5
Database Free Space : 320,477.281
Encryption Strength: AES

```

## Field descriptions

### Server Name

Specifies the name of the server.

### Server host name or IP address

Specifies the server TCP/IP address.

### Server TCP/IP port number

Specifies the server port address.

### Server URL

Specifies the URL address of the server that issued this command.

### Crossdefine

Specifies whether another server running the DEFINE SERVER command will automatically define itself to this server. See the SET CROSSDEFINE command.

## QUERY STATUS

### **Server Password Set**

Specifies whether the password has been set for the server.

### **Server Installation Date/Time**

Specifies the date and time when the server was installed.

### **Server Restart Date/Time**

Specifies the last date and time when 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.

### **Invalid Sign-on Attempt Limit**

Specifies the number of invalid sign-on attempts before a node is locked.

### **Minimum Password Length**

Specifies the minimum number of characters for the password.

### **Registration**

Specifies whether client node registration is open or closed.

### **Subfile Backup**

Specifies whether subfiles can be backed up to this server, as indicated by the SET SUBFILE command.

### **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.

### **Activity Log Retention**

Specifies the number of days information is retained in the activity log, or the size of the log.

### **Activity Summary Retention Period**

Specifies the number of days information is retained in the SQL activity summary table.

### **Activity Log Number of Records**

Specifies the number of records in the activity log.

### **Activity Log Size**

Specifies the size of the activity log.

### **License Audit Period**

Specifies the period, in days, after which the license manager automatically audits the Tivoli Storage Manager 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. Use the QUERY LICENSE command to see what factors are causing the server to fail to comply with the license terms.

**Central Scheduler**

Specifies whether central scheduling is running (active or inactive).

**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.

**Client Action Duration**

Specifies the duration of the period during which the client processes the schedule defined with the DEFINE CLIENTACTION command.

**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, 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, 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, the client node determines the number of minutes between retries.

**Scheduling Modes**

Specifies the central scheduling modes supported by the server.

**Active Receivers**

Specifies the receivers for which event logging has begun.

**Configuration manager?**

Specifies whether the server is a configuration manager.

**Refresh interval**

Specifies the interval that elapses before the managed server requests a refresh of any changes from a configuration manager.

**Last refresh date/time**

If the server is a managed server, specifies the date and time of the last successful refresh of configuration information from the configuration manager.

**Context Messaging**

Specifies whether context messaging is enabled or disabled. See the SET CONTEXTMESSAGING command.

**Table of Contents (TOC) Load Retention**

Displays the approximate number of minutes that unreferenced TOC data will be retained in the database.

## QUERY STATUS

### Machine Globally Unique ID

The globally unique identifier (GUID) as of the last time that the server was started. This GUID identifies the host machine on which the current server resides.

### Archive Retention Protection

Specifies if archive data retention protection is activated or deactivated.

### Database Directories

Specifies the locations of the database directories.

### Database Total Space

Specifies the total amount of space that is available, both used and free, in megabytes.

### Database Used Space

Specifies the total amount of space currently in use, in megabytes.

### Database Free Space

Specifies the space remaining in the file system where the path is located.

### Encryption Strength

Indicates data encryption strength: AES or DES.

## Related commands

Table 256. Commands related to QUERY STATUS

| Command              | Description  |
|----------------------|--|
| BEGIN EVENTLOGGING   | Starts event logging to a specified receiver.  |
| DISABLE SESSIONS     | Prevents new sessions from accessing IBM Tivoli Storage Manager but permits existing sessions to continue. |
| ENABLE SESSIONS      | Resumes server activity following the DISABLE command or the ACCEPT DATE command.                          |
| END EVENTLOGGING     | Ends event logging to a specified receiver.  |
| QUERY LICENSE        | Displays information about licenses and audits.  |
| SET ACCOUNTING       | Specifies whether accounting records are created at the end of each client session.                        |
| SET ACTLOGRETENTION  | Specifies the number of days to retain log records in the activity log.                                    |
| SET AUTHENTICATION   | Specifies whether clients are prompted for a password at the beginning of a session.                       |
| SET CONTEXTMESSAGING | Specifies to turn on context messaging to debug an ANR9999D message.                                       |
| SET CROSSDEFINE      | Specifies whether to cross define servers.   |
| SET EVENTRETENTION   | Specifies the number of days to retain records for scheduled operations.                                   |
| SET MAXCMDRETRIES    | Specifies the maximum number of retries after a failed attempt to execute a scheduled command.             |
| SET MAXSCHEDSESSIONS | Specifies the maximum number of client/server sessions available for processing scheduled work.            |



Table 256. Commands related to QUERY STATUS (continued)

| Command              | Description  |
|----------------------|--|
| SET PASSEXP          | Specifies the number of days after which a password is expired and must be changed.              |
| SET QUERYSCHEDPERIOD | Specifies the frequency for clients to obtain scheduled work, in client-polling mode.            |
| SET RANDOMIZE        | Specifies the randomization of start times within a window for schedules in client-polling mode. |
| SET REGISTRATION     | Specifies whether users can register themselves or must be registered by an administrator.       |
| SET RETRYPERIOD      | Specifies the time between retry attempts by the client scheduler.                               |
| SET SCHEDMODES       | Specifies the central scheduling mode for the server.  |
| SET SERVERHLADDRESS  | Specifies the high-level address of a server.  |
| SET SERVERLLADDRESS  | Specifies the low-level address of a server.   |
| SET SERVERNAME       | Specifies the name by which the server is identified.  |
| SET SERVERPASSWORD   | Specifies the server password.   |
| SET SUMMARYRETENTION | Specifies the number of days to retain information for the activity summary table.               |
| SET TOCLOADRETENTION | Specifies the number of minutes to retain information for unreferenced TOC sets.                 |

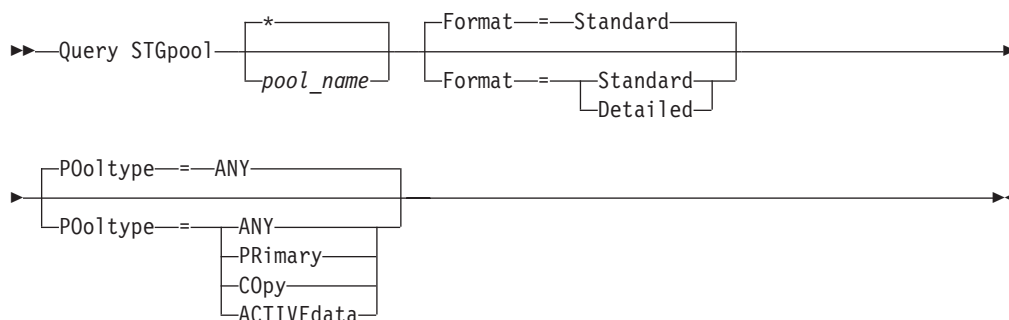
## QUERY STGPOOL (Query storage pools)

Use this 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

#### *pool\_name*

Specifies the storage pool to query. This parameter is optional. You can use wildcard characters to specify this name. If you do not specify a value for this parameter, all storage pools are displayed.

#### Format

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.

##### **Detailed**

Specifies that complete information is displayed.

#### P0oltype

Specifies the type of storage pool to query. This parameter is optional. The default value is `ANY`. Possible values are:

##### **ANY**

Query primary storage pools, copy storage pools, and active-data pools.

##### **PRimary**

Query only primary storage pools.

##### **COpy**

Query only copy storage pools.

##### **ACTIVEdata**

Query only active-data pools.

### Example: Display detailed random-access disk storage pool information

Display details for the storage pool named `DISKPOOL`. See “Field descriptions” on page 793 for field descriptions.

```
query stgpool diskpool format=detailed
```

```

Storage Pool Name: DISKPOOL
Storage Pool Type: Primary
Device Class Name: DISK
Estimated Capacity: 66 G
Space Trigger Util: 0.0
  Pct Util: 0.0
  Pct Migr: 3.1
  Pct Logical: 100.0
  High Mig Pct: 90
  Low Mig Pct: 70
  Migration Delay: 0
  Migration Continue: Yes
  Migration Processes: 1
  Reclamation Processes: 1
  Next Storage Pool:
  Reclaim Storage Pool:
Maximum Size Threshold: No Limit
  Access: Read/Write
  Description:
  Overflow Location:
  Cache Migrated Files?:
    Collocate?: Group
  Reclamation Threshold: 60
  Offsite Reclamation Limit:
Maximum Scratch Volumes Allowed: 32
  Number of Scratch Volumes Used: 1
  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
  Last Update by (administrator): SERVER_CONSOLE
  Last Update Date/Time: 02/24/2005 13:57:16
Storage Pool Data Format: Native
  Copy Storage Pool(s):
  Active-data Pool(s):
  Continue Copy on Error?: No
    CRC Data: Yes
  Reclamation Type: Threshold
  Overwrite Data when Deleted: 2 Time(s)

```

### Example: Display detailed sequential-access disk storage pool information

Display details for the storage pool named FILEPOOL. See “Field descriptions” on page 793 for field descriptions.

```
query stgpool filepool format=detailed
```

```

Storage Pool Name: FILEPOOL
Storage Pool Type: Primary
Device Class Name: FILEC
Estimated Capacity: 66 G
Space Trigger Util: 0.0
  Pct Util: 0.0
  Pct Migr: 3.1
  Pct Logical: 100.0
  High Mig Pct: 90
  Low Mig Pct: 70
  Migration Delay: 0
  Migration Continue: Yes
  Migration Processes: 1
  Reclamation Processes: 1
  Next Storage Pool:
  Reclaim Storage Pool:
Maximum Size Threshold: No Limit
  Access: Read/Write
  Description:
  Overflow Location:
  Cache Migrated Files?:
  Collocate?: Group
  Reclamation Threshold: 60
  Offsite Reclamation Limit:
Maximum Scratch Volumes Allowed: 32
Number of Scratch Volumes Used: 1
  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
  Last Update by (administrator): SERVER_CONSOLE
  Last Update Date/Time: 02/24/2005 13:57:16
Storage Pool Data Format: Native
  Copy Storage Pool(s):
  Active-data Pool(s):
  Continue Copy on Error?: No
  CRC Data: Yes
  Reclamation Type: Threshold
  Overwrite Data when Deleted: 2 Time(s)
  Deduplicate Data: Yes
Processes for Identifying Duplicates: 2
  Duplicate Data Not Stored: 1 GB

```

### Example: Display detailed sequential storage pool information

Display details for the active-data sequential storage pool named FILEPOOL which uses a FILE-type device class. See “Field descriptions” on page 793 for field descriptions.

```
query stgpool filepool format=detailed
```

```
Storage Pool Name: FILEPOOL
Storage Pool Type: Active-data
Device Class Name: FILEC
Estimated Capacity: 0.0 M
Space Trigger Util: 0.0
    Pct Util: 0.0
    Pct Migr: 0.0
    Pct Logical: 0.0
    High Mig Pct: 90
    Low Mig Pct: 70
    Migration Delay: 0
    Migration Continue: Yes
    Migration Processes: 1
    Reclamation Processes: 1
    Next Storage Pool:
    Reclaim Storage Pool:
    Maximum Size Threshold: No Limit
    Access: Read/Write
    Description:
    Overflow Location:
    Cache Migrated Files?:
    Collocate?: Group
    Reclamation Threshold: 60
    Offsite Reclamation Limit:
    Maximum Scratch Volumes Allowed: 99
    Number of Scratch Volumes Used: 0
    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
    Last Update by (administrator): SERVER_CONSOLE
    Last Update Date/Time: 05/03/2004 11:37:57
    Storage Pool Data Format: Native
    Copy Storage Pool(s):
    Active-data Pool(s):
    Continue Copy on Error:
    CRC Data: Yes
    Reclamation Type: Threshold
    Deduplicate Data: No
    Processes for Identifying Duplicates:
    Duplicate Data Not Stored:
```

**Example: Display summary information for a specific storage pool**

Display information for the storage pool named POOL1. See “Field descriptions” on page 793 for field descriptions.

query stgpool pool1

| Storage<br>Pool Name | Device<br>Class Name | Estimated<br>Capacity | Pct<br>Util | Pct<br>Migr | High<br>Mig<br>Pct | Low<br>Mig<br>Pct | Next<br>Storage<br>Pool |
|----------------------|----------------------|-----------------------|-------------|-------------|--------------------|-------------------|-------------------------|
| POOL1                | DISK                 | 58.5 M                | 0.8         | 0.7         | 90                 | 70                | POOL2                   |

**Example: Display detailed 8 mm tape storage pool information**

Display details for the storage pool named 8MMPOOL. See “Field descriptions” on page 793 for field descriptions.

query stgpool 8mmpool format=detailed

```

Storage Pool Name: 8MMP00L
Storage Pool Type: Primary
Device Class Name: 8MMTAPE
Estimated Capacity: 0.0 M
Space Trigger Util: 0.0
Pct Util: 0.0
Pct Migr:
Pct Logical: 0.0
High Mig Pct: 90
Low Mig Pct: 70
Migration Delay: 0
Migration Continue: Yes
Migration Processes: 1
Reclamation Processes: 1
Next Storage Pool:
Reclaim Storage Pool:
Maximum Size Threshold: 5 M
Access: Read/Write
Description: Main storage pool
Overflow Location: Room1234/Bldg31
Cache Migrated Files?:
Collocate?: No
Reclamation Threshold: 60
Offsite Reclamation Limit:
Maximum Scratch Volumes Allowed: 5
Number of Scratch Volumes Used: 3
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
Last Update by (administrator): ADMIN
Last Update Date/Time: 03/22/2002 06:55:45
Storage Pool Data Format: Native
Copy Storage Pool(s): COPYPOOL1
Active-data Pools: ACTIVEPOOL1 ACTIVEPOOL2
Continue Copy on Error: Yes
CRC Data: Yes
Reclamation Type: Threshold

```

### Example: Display detailed NAS2CLASS storage pool information

Display details for a storage pool, NAS2LIBPOOL. When you set up this storage pool, you set the data format to NETAPPDUMP. See “Field descriptions” on page 793 for field descriptions.

```
query stgpool nas2libpool format=detailed
```

```

Storage Pool Name: NAS2
Storage Pool Name: NAS2LIBPOOL
Storage Pool Type: Primary
Device Class Name: NAS2CLASS
Estimated Capacity: 0.0 M
Space Trigger Util:
    Pct Util: 0.0
    Pct Migr:
    Pct Logical: 0.0
    High Mig Pct:
    Low Mig Pct:
    Migration Delay:
    Migration Continue:
    Migration Processes:
    Reclamation Processes:
    Next Storage Pool:
    Reclaim Storage Pool:
    Maximum Size Threshold:
        Access: Read/Write
        Description:
        Overflow Location:
    Cache Migrated Files?:
        Collocate?: Group
    Reclamation Threshold:
    Offsite Reclamation Limit:
    Maximum Scratch Volumes Allowed: 50
    Number of Scratch Volumes Used: 0
    Delay Period for Volume Reuse: 0 Day(s)
    Migration in Progress?:
        Amount Migrated (MB):
    Elapsed Migration Time (seconds):
    Reclamation in Progress?:
    Last Update by (administrator): SERVER_CONSOLE
    Last Update Date/Time: 07/08/2004 16:24:43
    Storage Pool Data Format: NetApp Dump
    Copy Storage Pool(s):
    Active-data Pool(s):
    Continue Copy on Error?: No
        CRC Data: No
    Reclamation Type:

```

## Field descriptions

### Storage Pool Name

The name of the storage pool.

### Storage Pool Type

The type of storage pool.

### Device Class Name

The name of the device class assigned to the storage pool.

### Estimated Capacity

The estimated capacity of the storage pool in megabytes (M) or gigabytes (G).

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 of the sequential access volumes in the storage pool, regardless of their access mode. When at least one volume is used (including both scratch and private volumes), the estimated capacity includes the unused number of scratch volumes allowed for the pool and the number of empty private volumes that you have defined to the pool.

For Centera, this represents the total capacity of the Centera storage device being queried.

### Space Trigger Util

Utilization of the storage pool, as calculated by the storage pool space trigger, if any, for this storage pool. Note that you can define space triggers for storage pools associated with DISK or FILE device types only.

For sequential access devices, space trigger utilization is expressed as a percentage of the number of used bytes on each sequential access volume relative to the size of the volume, and the estimated capacity of all existing volumes in the storage pool. It does not include potential scratch volumes. Unlike the calculation for percent utilization (Pct Util), the calculation for space trigger utilization favors creation of new private file volumes by the space trigger over usage of additional scratch volumes.

For disk devices, space trigger utilization is expressed as a percentage of the estimated capacity, including cached data. However, it excludes data that resides on any volumes that are varied offline. The value for space trigger utilization can be higher than the value for percent migration (Pct Migr) if you issue QUERY STGPOOL while a file creation is in progress. The value for space trigger utilization is determined by the amount of space actually allocated while the transaction is in progress. The value for percent migration represents only the space occupied by committed files. At the end of the transaction, these values are synchronized.

The value for space trigger utilization includes cached data on disk volumes. Therefore, when cache is enabled and migration occurs, the value remains the same because the migrated data remains on the volume as cached data. The value decreases only when the cached data expires or when the space that cached files occupy needs to be used for no-cached files.

### Pct Util

An estimate of the utilization of the storage pool, as a percentage.

For sequential access devices, this is expressed 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, including the number of potential scratch volumes that may be allocated.

For disk devices, this is expressed as a percentage of the estimated capacity, including cached data and data that resides on any volumes that are varied offline. The value for Pct Util can be higher than the value for Pct Migr if you issue this command while a file creation transaction is in progress. The value for Pct Util is determined by the amount of space actually allocated (while the transaction is in progress). The value for Pct Migr represents only the space occupied by committed files. At the end of the transaction, these values become synchronized.

The Pct Util value includes cached data on disk volumes. Therefore, when cache is enabled and migration occurs, the Pct Util value remains the same because the migrated data remains on the volume as cached data. The Pct Util value decreases only when the cached data expires or when the space that cached files occupy needs to be used for noncached files.

For Centera, this represents an estimate of the utilization of the entire Centera storage device, not the storage pool being queried.

### Pct Migr (*primary storage pools only*)

An estimate of the percentage of data in the storage pool that can be migrated. The server uses this value and the high and low migration thresholds to determine when to start and stop migration.



For random-access 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 disk devices, this value is specified as a percentage of the value for the estimated capacity, including the capacity of all the scratch volumes specified for the pool. For other types of sequential-access devices, this value is the percentage of the total number of volumes in the pool that contain at least one byte of active data. The total number of volumes includes the maximum number of scratch volumes.

The Pct Util value includes cached data on a volume; the Pct Migr value excludes cached data. Therefore, when cache is enabled and migration occurs, the Pct Migr value decreases but the Pct Util value remains the same because the migrated data remains on the volume as cached data. The Pct Util value decreases only when the cached data expires or when the space that cached files occupy needs to be used for noncached files.

#### **Pct Logical**

The logical occupancy of the storage pool as a percentage of the total occupancy. Logical occupancy is space occupied by client files that may or may not be part of an aggregate. A Pct Logical value less than 100% indicates that there is vacant space within aggregates in the storage pool.

#### **High Mig Pct** (*primary storage pools only*)

The high migration threshold, which specifies when the server can begin migration for the storage pool. The server starts migration processes when capacity utilization reaches this threshold.

#### **Low Mig Pct** (*primary storage pools only*)

The low migration threshold, which specifies when the server can stop migration for the storage pool. The server stops migration processes when capacity utilization reaches this threshold.

#### **Migration Delay** (*primary storage pools only*)

The minimum number of days that a file must remain in a storage pool before the server can migrate the file to the next storage pool. For a disk storage pool, the days are counted from the time that the file was stored in the storage pool or last retrieved by a client. For a sequential access storage pool, the days are counted from the time that the file was stored in the storage pool.

#### **Migration Continue** (*primary storage pools only*)

Whether the server continues to migrate files to the next storage pool even if the files have not been in the pool for the number of days specified by the migration delay.

#### **Migration Processes**

The number of parallel processes that are used for migrating files from a random or sequential access primary storage pool.

#### **Reclamation Processes**

The number of parallel processes that are used for reclaiming the volumes in a sequential access primary or copy storage pool.

#### **Next Storage Pool** (*primary storage pools only*)

The storage pool that is the destination for data that is migrated from this storage pool.

#### **Reclaim Storage Pool** (*primary, sequential access storage pools only*)

If specified, the storage pool that is the destination for data moved from

volumes during reclamation processing. If no pool is specified, by default reclamation processing moves data only among volumes within the same storage pool.

**Maximum Size Threshold** (*primary storage pools only*)

The maximum size of files that may be stored in the storage pool.

**Access**

How the data in the storage pool can be accessed.

**Description**

The description of the storage pool.

**Overflow Location** (*sequential access storage pools only*)

The location where volumes in the storage pool are stored when they are ejected from an automated library with the MOVE MEDIA command.

**Cache Migrated Files?** (*random access storage pools only*)

Whether caching is enabled for files migrated to the next storage pool.

**Collocate?** (*sequential access storage pools only*)

Whether collocation is disabled or enabled. If collocation is disabled, the value of this field is No. If collocation is enabled, the possible values are Group, Node, and Filespace.

**Reclamation Threshold** (*sequential access storage pools only*)

The threshold that determines when volumes in a storage pool are reclaimed. The server compares the percentage of reclaimable space on a volume to this value to determine if reclamation is necessary.

**Offsite Reclamation Limit**

The number of offsite volumes having their space reclaimed during reclamation for this storage pool. This field applies only when POOLTYPE=COPY.

**Maximum Scratch Volumes Allowed** (*sequential access storage pools only*)

The maximum number of scratch volumes that the server can request for the storage pool.

**Number of Scratch Volumes Used** (*sequential access storage pools only*)

The number of scratch volumes used in the storage pool.

**Delay Period for Volume Reuse** (*sequential access storage pools only*)

The number of days that must elapse after all files have been deleted from a volume, before the server returns the volume to scratch or reuses the volume.

**Migration in Progress?** (*primary storage pools only*)

Whether at least one migration process is active for the storage pool.

**Amount Migrated (MB)** (*primary storage pools only*)

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)** (*primary storage pools only*)

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?** (*sequential access storage pools only*)

Whether a reclamation process is active for the storage pool.

**Last Update by (administrator)**

The name of the administrator that has defined or most recently updated the storage pool.

**Last Update Date/Time**

The date and time that an administrator defined or most recently updated the storage pool.

**Storage Pool Data Format**

The type of data format used to write data to this storage pool (for example NATIVE, NETAPPDUMP, CELERRADUMP, or NDMPDUMP).

**Copy Storage Pool (s)**

The copy storage pools listed here will have data simultaneously written to them when data is backed up or archived to the primary storage pool queried by this command.

**Active-data Pool(s)**

The active-data pools listed here will have data simultaneously written to them when data is backed up to the primary storage pool queried by this command.

**Continue Copy on Error?**

Whether a server should continue writing data to other copy storage pools in the list or terminate the entire transaction when a write failure occurs to one of the copy pools in the list. This field applies only to primary random-access and primary sequential-access storage pools.

**CRC Data**

Whether data is validated by a cyclic redundancy check (CRC) when data is transferred during data storage and retrieval on a device.

**Reclamation Type**

Whether volumes in this storage pool are reclaimed by threshold or by SnapLock retention date.

**Overwrite Data when Deleted**

The number of times data will be physically overwritten after it has been deleted from the database.

**Deduplicate Data**

Whether data in the storage pool will be deduplicated.

**Processes for Identifying Duplicates**

The number of duplicate-identification processes that are specified as the default for the storage pool. The number of duplicate-identification processes that are specified in this field might not equal the number of duplicate-identification processes that are running.

**Duplicate Data Not Stored**

Represents the amount of data that was removed from the storage pool by reclamation processing and the amount of storage space that was saved in this storage pool as a result of deduplication.

### Related commands

*Table 257. Commands related to QUERY STGPOOL*

| Command           | Description   |
|-------------------|---|
| COPY ACTIVATEDATA | Copies active backup data.  |
| DEFINE STGPOOL    | Defines a storage pool as a named collection of server storage media. |
| DELETE STGPOOL    | Deletes a storage pool from server storage.                           |
| UPDATE STGPOOL    | Changes the attributes of a storage pool.                             |

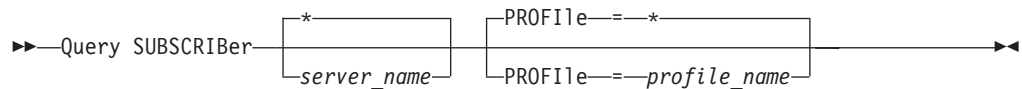
## QUERY SUBSCRIBER (Display subscriber information)

Use this command on a configuration manager to display information about subscribers and their profile subscriptions.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *server\_name*

Specifies the name of managed server a for which subscription information is displayed. You can use wildcard characters to specify multiple server names. This parameter is optional. The default is all managed servers.

#### PROFILE

Specifies a profile name for which information is displayed. You can use wildcard characters to specify multiple profile names. This parameter is optional. The default is all profiles.

### Example: List a configuration manager's profile subscriptions

Display subscriber information for all profile subscriptions to this configuration manager. See "Field descriptions" for field descriptions.

query subscriber

| Subscriber | Profile name    | Is current? | Last update date/time         |
|------------|-----------------|-------------|-------------------------------|
| SERVER2    | DEFAULT_PROFILE | Yes         | Thu, May 14, 1998 01:14:42 PM |
| SERVER2    | SETUP           | Yes         | Thu, May 14, 1998 01:14:42 PM |

### Field descriptions

#### Subscriber

The name of the subscriber (managed server).

#### Profile name

The name of the profile.

#### Is current?

Whether the subscription has been refreshed with the current information associated with the profile. Possible values are:

**Yes** The managed server is current.

**No** The managed server is not current. If this field is NO after the profile has been refreshed, check the server messages for error conditions that might cause the refresh to fail.

## QUERY SUBSCRIBER

### Unknown

Either the managed server has a more recent version of the profile than the configuration manager, or the profile no longer exists on the configuration manager, but the subscription is still associated with the profile.

### Last update date/time

Specifies the date and time that configuration information for the subscription was successfully distributed to the subscriber.

## Related commands

*Table 258. Commands related to QUERY SUBSCRIBER*

| Command             | Description  |
|---------------------|--|
| DEFINE SUBSCRIPTION | Subscribes a managed server to a profile.                    |
| DELETE SUBSCRIBER   | Deletes obsolete managed server subscriptions.               |
| DELETE SUBSCRIPTION | Deletes a specified profile subscription.                    |
| NOTIFY SUBSCRIBERS  | Notifies servers to refresh their configuration information. |
| SET CONFIGMANAGER   | Specifies whether a server is a configuration manager.       |
| QUERY SUBSCRIPTION  | Displays information about profile subscriptions.            |

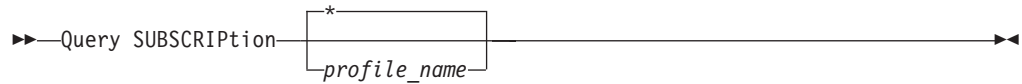
## QUERY SUBSCRIPTION (Display subscription information)

Use this command on a managed server to display profile subscription information.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *profile\_name*

Specifies the name of the profile for which subscription information is displayed. You can use wildcard characters to specify multiple names. This parameter is optional. The default is all profiles.

### Example: Display description information

Display subscription information for all profiles.

```
query subscription
```

| Configuration manager | Profile name    | Last update date/time         |
|-----------------------|-----------------|-------------------------------|
| -----                 | -----           | -----                         |
| SERVER1               | ADMIN_INFO      | Thu, May 14, 1998 01:35:13 PM |
| SERVER1               | DEFAULT_PROFILE | Thu, May 14, 1998 01:35:13 PM |
| SERVER1               | EMPLOYEE        | Thu, May 14, 1998 01:35:13 PM |

### Field descriptions

#### Configuration manager

The name of the configuration manager.

#### Profile name

The name of the profile.

#### Last update date/time

When the most recent configuration information was successfully distributed to the subscriber.

### Related commands

Table 259. Commands related to QUERY SUBSCRIPTION

| Command             | Description                                    |
|---------------------|--|
| DEFINE SUBSCRIPTION | Subscribes a managed server to a profile.      |
| DELETE SUBSCRIBER   | Deletes obsolete managed server subscriptions. |
| DELETE SUBSCRIPTION | Deletes a specified profile subscription.      |

## QUERY SUBSCRIPTION

*Table 259. Commands related to QUERY SUBSCRIPTION (continued)*

| Command            | Description   |
|--------------------|---|
| NOTIFY SUBSCRIBERS | Notifies servers to refresh their configuration information.                |
| QUERY SUBSCRIBER   | Displays information about subscribers and their subscriptions to profiles. |



## QUERY SYSTEM (Query the system configuration and capacity)

Use this command to obtain consolidated information about the server's configuration and capacity.

This command consolidates output from the following commands:

|                   |                  |
|-------------------|------------------|
| QUERY ASSOCIATION | QUERY OPTION     |
| QUERY COPYGROUP   | QUERY PROCESS    |
| QUERY DB          | QUERY SCHEDULE   |
| QUERY DBSPACE     | QUERY SESSION    |
| QUERY DEVCLASS    | QUERY STATUS     |
| QUERY DOMAIN      | QUERY STGPOL     |
| QUERY LOG         | QUERY VOLHISTORY |
| QUERY MGMTCLASS   | QUERY VOLUME     |

### Privilege class

Any administrator can issue this command.

### Syntax

►►—Query SYStem—◄◄

### Example: Display consolidated system information

Issue the QUERY SYSTEM command to obtain consolidated system information. For sample outputs for these query commands, see the individual commands.

```
query system
```

### Related commands

Table 260. Commands related to QUERY SYSTEM

| Command           | Description   |
|-------------------|---|
| QUERY ASSOCIATION | Displays the clients associated with one or more schedules. |
| QUERY COPYGROUP   | Displays the attributes of a copy group.                    |
| QUERY DB          | Displays allocation information about the database.         |
| QUERY DEVCLASS    | Displays information about device classes.                  |
| QUERY DOMAIN      | Displays information about policy domains.                  |
| QUERY LOG         | Displays information about the recovery log.                |
| QUERY MGMTCLASS   | Displays information about management classes.              |
| QUERY OPTION      | Displays information about server options.                  |
| QUERY PROCESS     | Displays information about background processes.            |
| QUERY SCHEDULE    | Displays information about schedules.                       |

## QUERY SYSTEM

*Table 260. Commands related to QUERY SYSTEM (continued)*

| Command       | Description  |
|---------------|--|
| QUERY SESSION | Displays information about all active administrator and client sessions with IBM Tivoli Storage Manager. |
| QUERY STATUS  | Displays the settings of server parameters, such as those selected by the SET commands.                  |
| QUERY STGPOOL | Displays information about storage pools.  |
| QUERY VOLUME  | Displays information about storage pool volumes.   |

## QUERY TAPEALERTMSG (Display status of SET TAPEALERTMSG command)

Use this command to display the status of the SET TAPEALERTMSG command. You can enable or disable tape alerts. When enabled, Tivoli Storage Manager can retrieve diagnostic information from a tape or library device and display it using ANR messages. When disabled, Tivoli Storage Manager will not query a device for this information.

### Privilege class

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

### Syntax

►►—Query TAPEAlertmsg—◄◄

### Example: Display the status of the QUERY TAPEALERTMSG command

Use the QUERY TAPEALERTMSG command to determine if tape alerts are to be retrieved from devices and displayed in the form of ANR messages.

```
query tapealertmsg
```

```
ANR2017I Administrator SERVER_CONSOLE issued command:
      QUERY TAPEALERTMSG
ANR8960I QUERY TAPEALERTMSG: The display of Tape Alerts from SCSI
      devices is Enabled.
```

### Related commands

Table 261. Commands related to QUERY TAPEALERTMSG

| Command          | Description   |
|------------------|---|
| SET TAPEALERTMSG | Specifies whether tape and library devices report diagnostic information to the server. |

# QUERY TOC (Display table of contents for a backup image)

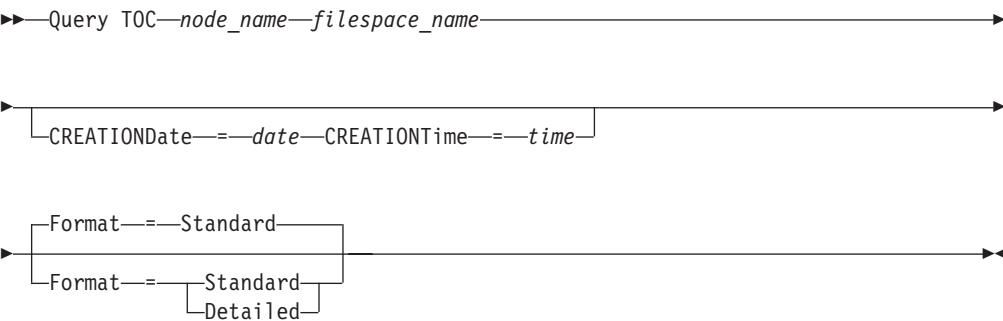
Use this command to display directory and file information contained in the table of contents (TOC) for a specified backup image. This command will not load table of contents information into the IBM Tivoli Storage Manager database. The specified table of contents will be read from a storage pool each time the QUERY TOC command is issued.

This command cannot be issued from the server console. If the table of contents is stored on removable media, a mount point will be required and output will be delayed while the storage pool volume is mounted.

## Privilege class

To issue this command you must have either system privilege, policy privilege for the domain to which the node is assigned, or client owner authority over the node.

## Syntax



## Parameters

### node\_name (Required)

Specifies the name of the NAS node to which the table of contents (TOC) belongs. You cannot use wildcards to specify this name.

### filespace\_name (Required)

Specifies the name of the file space to which the table of contents belongs. The file space name you specify cannot contain wildcard characters.

### CREATIONDate

Specifies the creation date of the backup image for which the table of contents is to be displayed. This parameter is optional. If you specify CREATIONDate, you must also specify CREATIONTIME. If you do not specify these parameters, the contents of the latest backup image for the specified node and file space will be displayed, provided that this image has a table of contents. You can only specify the creation date as the following:

| Value      | Description     | Example    |
|------------|-----------------|------------|
| MM/DD/YYYY | A specific date | 05/15/2002 |

This specifies that you want to display the contents of the backup image created on this date. You can obtain this date from the output of the QUERY NASBACKUP command.

### CREATIONTime

Specifies the creation time of the backup image for which the table of contents is to be displayed. This parameter is optional. If you specify CREATIONTIME,

you must also specify CREATIONDate. If you do not specify these parameters, the contents of the latest backup image for the specified node and file space will be displayed, provided that this image has a table of contents. You can only specify the creation time as the following:

| Value    | Description                                     | Example  |
|----------|---|----------|
| HH:MM:SS | A specific time on the specified creation date. | 10:30:08 |

This specifies that you want to display the contents of the backup image created on this time for the specified date. You can obtain this time from the output of the QUERY NASBACKUP command.

### Format

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 files.

#### Detailed

Specifies that complete information is displayed for the files, including the hexadecimal representation of each file or directory name.

### Example: Display detailed table of contents information for a specific node

Use the QUERY TOC command to display information in the table of contents belonging to NAS node NETAPP in the file space /vol/vol1 created on 12/06/2002 at 11:22:46. Specify a detailed format.

```
query toc netapp /vol/vol1 creationdate=12/06/2002 creationtime=11:22:46
format=detailed
```

```
Objects in the image backed up on 12/06/2002 11:22:46
for filespace /vol/vol1 in node NETAPP:

                                Object Name: /.etc
Hexadecimal Object Name: 2f657463
                                Object Type: Directory
                                Object Size: 4,096
Last data Modification Date/Time: 07/31/2002 14:21:19

                                Object Name: /.etc/oldmaps/ndmp
Hexadecimal Object Name: 2f6574632f6f6c646d6170
                                732f6e646d70
                                Object Type: Directory
                                Object Size: 4,096
Last data Modification Date/Time: 07/31/2002 14:21:19

                                Object Name: /.etc/oldmaps/ndmp/TSM
                                /vol/vol1/3df0e8fd
Hexadecimal Object Name: 2f6574632f6f6c646d6170
                                732f6e646d702f54534d2
                                02f766f6c2f766f6c312f3
                                364663065386664
                                Object Type: File
                                Object Size: 36,864
Last data Modification Date/Time: 12/06/2002 11:14:22
```

### Related commands

*Table 262. Commands related to QUERY TOC*

| Command         | Description                                     |
|-----------------|---|
| BACKUP NODE     | Backs up a network-attached storage (NAS) node. |
| QUERY NASBACKUP | Displays information about NAS backup images.   |
| RESTORE NODE    | Restores a network-attached storage (NAS) node. |

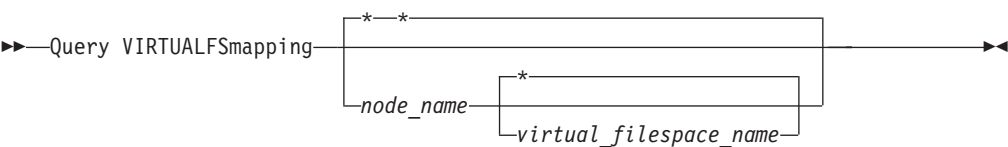
## QUERY VIRTUALFSMAPPING (Query a virtual file space mapping)

Use this command to query a virtual file space mapping definition.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *node\_name*

Specifies the client node to which the virtual file space belongs. You can use wildcard characters to specify this name. This parameter is optional. The default is all client node names. You must specify a value for this parameter if you specify a virtual file space name.

#### *virtual\_filespace\_name*

Specifies the name of the virtual file space mappings to be queried. You can use wildcard characters to specify this name. This parameter is optional. If a value is not specified, all virtual file space mappings are queried. Virtual file space mapping names are case sensitive. Use the QUERY VIRTUALFSMAPPING command to determine the correct capitalization for the virtual file space mapping to be queried.

### Example: Display virtual file spaces for a specific node

Display the currently defined virtual file spaces for node NAS1. See “Field descriptions” for field descriptions.

```
query virtualfsmapping nas1
```

| Node Name | Virtual Filespace Mapping Name | Filespace Name | Path             | Hexadecimal Path? |
|-----------|--------------------------------|----------------|------------------|-------------------|
| NAS1      | /mikesdir                      | /vol/vol2      | /mikes           | No                |
| NAS1      | /tmpdir                        | /vol/vol1      | /tmp             | No                |
| NAS1      | /nonASCIIIDir                  | /vol/vol3      | 2f73657276657231 | Yes               |

### Field descriptions

#### **Node Name**

Specifies the name of the client node.

#### **Virtual Filespace Mapping Name**

Specifies the name of the virtual file space mapping.

#### **Filespace Name**

Specifies the name of the file space for the client node.

File space names and file names that can be in a different code page or locale than the server do not display correctly on the Administration

## QUERY VIRTUALFSMAPPING

Center or the administrative command-line interface. The data itself is backed up and can be restored properly, but the file space or file name may display with a combination of invalid characters or blank spaces.

If the file space name is Unicode enabled, the name is converted to the server's code page for display. The results of the conversion for characters not supported by the current code page depends on the operating system. For names that Tivoli Storage Manager is able to partially convert, you might see question marks (??), blanks, unprintable characters, or "...". These characters indicate to the administrator that files do exist. If the conversion is not successful, the name is displayed as "...". Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

**Path** Specifies the path to the client node.

### Hexadecimal Path

Indicates whether the path is hexadecimal.

## Related commands

*Table 263. Commands related to QUERY VIRTUALFSMAPPING*

| Command                 | Description                          |
|-------------------------|--------------------------------------|
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping. |
| DELETE VIRTUALFSMAPPING | Delete a virtual file space mapping. |
| UPDATE VIRTUALFSMAPPING | Update a virtual file space mapping. |



## QUERY VOLHISTORY (Display sequential volume history information)

Use this command to display sequential volume history information. To save sequential volume history information to one or more files, use the BACKUP VOLHISTORY command.

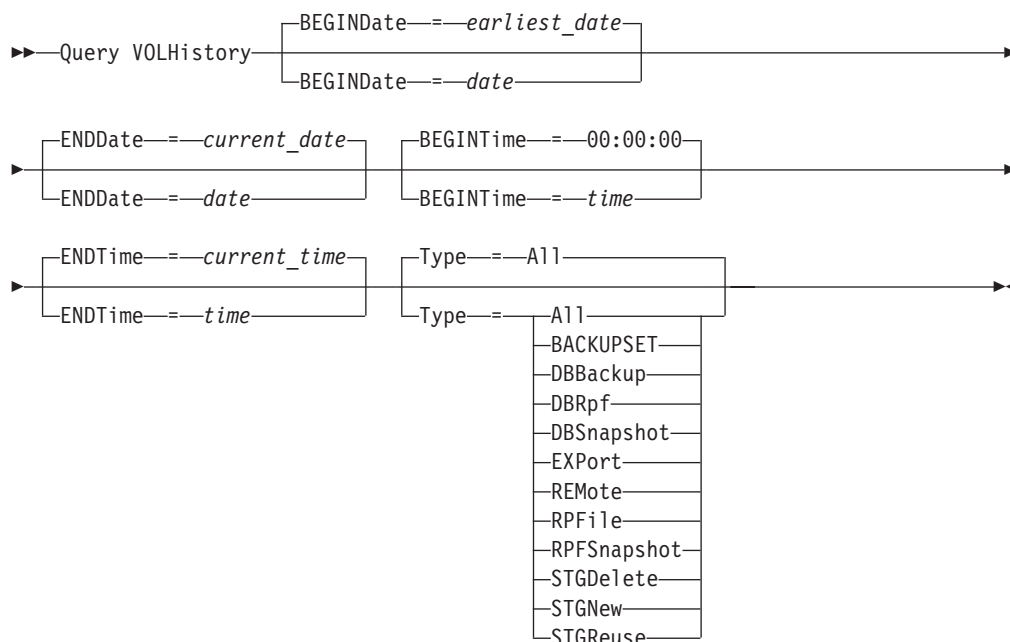
Use the VOLUMEHISTORY server option to specify one or more volume history files. After the server is restarted, Tivoli Storage Manager updates volume information in both the database and the files.

Use the QUERY BACKUPSET command to query specified backup set information.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### BEGINDate

Specifies that you want to display information beginning with records created on the specified date. This parameter is optional. The default is the earliest date for which history information exists.

You can specify the date using one of the values below:

| Value      | Description      | Example    |
|------------|------------------|------------|
| MM/DD/YYYY | A specific date  | 09/15/1998 |
| TODAY      | The current date | TODAY      |

## QUERY VOLHISTORY

| Value               | Description                           | Example  |
|---------------------|---------------------------------------|--|
| TODAY-days or -days | The current date minus days specified | TODAY-7 or -7.<br><br>To display information beginning with records created a week ago, specify BEGINDATE=TODAY-7 or BEGINDATE=-7. |

### ENDDate

Specifies that you want to display information ending with records created on the specified date. This parameter is optional. The default is the current date.

You can specify the date using one of the values below:

| Value               | Description  | Example  |
|---------------------|--|--|
| MM/DD/YYYY          | A specific date  | 09/15/1998   |
| TODAY               | The current date   | TODAY  |
| TODAY-days or -days | The current date minus days specified. The maximum number of days is 9999. | TODAY-1 or -1.<br><br>To display records created up to yesterday, specify ENDDATE=TODAY-1 or ENDDATE=-1. |

### BEGINTime

Specifies that you want to display information beginning with records created at the specified time. This parameter is optional. The default is midnight (00:00:00).

You can specify the time using one of the values below:

| Value               | Description  | Example  |
|---------------------|--|--|
| HH:MM:SS            | A specific time on the specified begin date                          | 12:33:28   |
| NOW                 | The current time on the specified begin date                         | NOW  |
| NOW+HH:MM or +HH:MM | The current time plus hours and minutes on the specified begin date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 with BEGINTIME=NOW+03:00 or BEGINTIME=+03:00, Tivoli Storage Manager displays records with a time of 12:00 or later on the begin date. |
| NOW-HH:MM or -HH:MM | The current time minus hours and minutes on the specified begin date | NOW-03:30 or -03:30.<br><br>If you issue this command at 9:00 with BEGINTIME=NOW-03:30 or BEGINTIME=-03:30, Tivoli Storage Manager displays records with a time of 5:30 or later on the begin date.  |

### ENDTime

Specifies that you want to display information ending with records created at the specified time on the end date. This parameter is optional. The default is the current time.

You can specify the time using one of the values below:

| Value                                  | Description  | Example  |
|--|--|--|
| <i>HH:MM:SS</i>                        | A specific time on the specified end date                          | 10:30:08   |
| NOW                                    | The current time on the specified end date                         | NOW  |
| NOW+ <i>HH:MM</i> or<br>+ <i>HH:MM</i> | The current time plus hours and minutes on the specified end date  | NOW+03:00 or +03:00.<br><br>If you issue this command at 9:00 with<br>ENDTIME=NOW+03:00 or<br>ENDTIME=+03:00, Tivoli Storage<br>Manager displays records with a time<br>of 12:00 or later on the end date. |
| NOW- <i>HH:MM</i> or<br>- <i>HH:MM</i> | The current time minus hours and minutes on the specified end date | NOW-03:30 or -03:30<br><br>If you issue this command at 9:00 with<br>ENDTIME=NOW-3:30 or<br>ENDTIME=-3:30, Tivoli Storage<br>Manager displays records with a time<br>of 5:30 or earlier on the end date.   |

### Type

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 all records.

#### BACKUPSET

Specifies to display only information about backup set volumes.

#### DBBackup

Specifies to display only records that contain information about full and incremental database backup volumes, that is with the volume types of BACKUPFULL and BACKUPINCR.

#### DBRpf

Specifies to display only records that contain information about full and incremental database backup volumes and recovery plan file object volumes (volume types of BACKUPFULL, BACKUPINCR, and RPFIL).

#### DBSnapshot

Specifies to display only records that contain information about volumes used for database snapshot backups.

#### EXPort

Specifies only records that contain information about export volumes.

#### REMote

Specifies to display only records that contain information about volumes used by library clients.

#### RPFil

Specifies to display only records that contain information about recovery plan file objects that were created assuming database full and incremental backups and that are saved on a target server.

#### RPFSnapshot

Specifies to display only records that contain information about snapshot

## QUERY VOLHISTORY

recovery plan file objects that were created assuming snapshot database backups and that are saved on a target server.

### **STGDelete**

Specifies only records that contain information about deleted sequential storage pool volumes.

### **STGNew**

Specifies only records that contain information about new sequential access storage volumes.

### **STGReuse**

Specifies only records that contain information about reused sequential storage pool volumes.

## **Example: Display volume history information**

Display volume history information stored in the Tivoli Storage Manager database. See “Field descriptions” for field descriptions. Issue the command:

```
query volhistory
```

```
Date/Time: 02/03/2009 18:28:06
Volume Type: STGNEW
Backup Series: 0
Backup Operation: 0
Volume Seq: 0
Device Class: FILE
Volume Name: /adsmfct/server/prvol1
Volume Location:
Command:
```

**Note:** The volume history file will contain additional fields that do not appear in the query output. These fields are specific to database backup and restore support. They are not intended for use or modification by Tivoli Storage Manager administrators. The fields will be bracketed with a message indicating these are for Tivoli Storage Manager internal use only and not meant to be modified.

## **Field descriptions**

### **Date/Time**

The date and time that the volume was created.

### **Volume Type**

The type of volume:

#### **BACKUPFULL**

Full database backup volume.

#### **BACKUPINCR**

Incremental database backup volume.

#### **BACKUPSET**

Client backup set volume.

#### **DBSNAPSHOT**

Snapshot database backup volume.

#### **EXPORT**

Export volume.

#### **REMOTE**

A volume used on the library client, which is the Tivoli Storage

Manager server named in the Volume Location field. See the volume history on the server that is the library client to get details about how the volume is used.

**RPFIL**

Recovery plan file object volume created assuming full and incremental database backups.

**RPFSSnapshot**

Recovery plan file object volume created assuming snapshot database backups.

**STGDELETE**

Deleted sequential access storage pool volume.

**STGNEW**

Added sequential access storage pool volume.

**STGREUSE**

Reused sequential access storage pool volume.

**Backup Series**

The value of this field depends on the volume type:

- For BACKUPFULL or BACKUPINCR volume types: the backup series identifier.
- For the DBSNAPSHOT volume type: the identifier of the backup series that is associated with the DBSNAPSHOT entry.
- For the RPFIL volume type: the identifier of the backup series that is associated with the RPFIL entry.
- For the RPFSSNAPSHOT volume type: the identifier of the backup series that is associated with the RPFSSNAPSHOT entry.
- For BACKUPSET volume types: this field is blank.
- For all other volume types: always 0.

A backup series is a full backup and all incremental backups that apply to that full backup. Another series begins with the next full backup of the database.

**Backup Operation**

For BACKUPFULL or BACKUPINCR volume types: 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.

For DBSNAPSHOT volume types: the operation number of this DBSNAPSHOT volume within the DBSNAPSHOT series.

For all other volume types: always 0.

This field is blank when the volume type is BACKUPSET.

**Volume Seq**

The sequence or position of the volume within the backup series.

- For BACKUPFULL or BACKUPINCR volume types: 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.

- For BACKUPSET volume types: the sequence, or position, of the volume within the BACKUPSET series.
- For DBSNAPSHOT volume types: the sequence, or position, of the volume within the DBSNAPSHOT series. Volume sequence 1 identifies the first volume used for the first DBSNAPSHOT operation, and so on.
- For EXPORT volume types: the sequence number of the volume when it was used for exporting data.
- For RPFIL volume types: the value of this field is always one (1).
- For all other volume types: always 0.

### Device Class

The name of the device class associated with this volume.

### Volume Name

The name of the volume.

### Volume Location

The location of the volume. This information is available only for the following volume types:

BACKUPFULL  
BACKUPINCR  
EXPORT  
REMOTE  
RPFIL

For the volume type of REMOTE, this location field is the server name of the library client that owns this volume.

For the volume type of RPFIL, this location field is the server name defined in the device class definition used by the PREPARE command when the DEVCLASS parameter is specified.

### Command

When the volume type is EXPORT or BACKUPSET and the volume sequence is 1 (for example, the first volume), this field shows the command that was used to generate the volume. If the EXPORT or BACKUPSET is on more than one volume, the command is displayed with the first volume but not with any of the other volumes.

For any volume type other than EXPORT or BACKUPSET, this field is blank.

## Related commands

Table 264. Commands related to QUERY VOLHISTORY

| Command           | Description   |
|-------------------|---|
| BACKUP VOLHISTORY | Records volume history information in external files.                       |
| DELETE VOLHISTORY | Removes sequential volume history information from the volume history file. |
| PREPARE           | Creates a recovery plan file.   |
| QUERY RPFIL       | Displays information about recovery plan files.                             |

*Table 264. Commands related to QUERY VOLHISTORY (continued)*

| Command           | Description   |
|-------------------|---|
| QUERY BACKUPSET   | Displays backup sets.   |
| UPDATE VOLHISTORY | Adds or changes location information for a volume in the volume history file. |

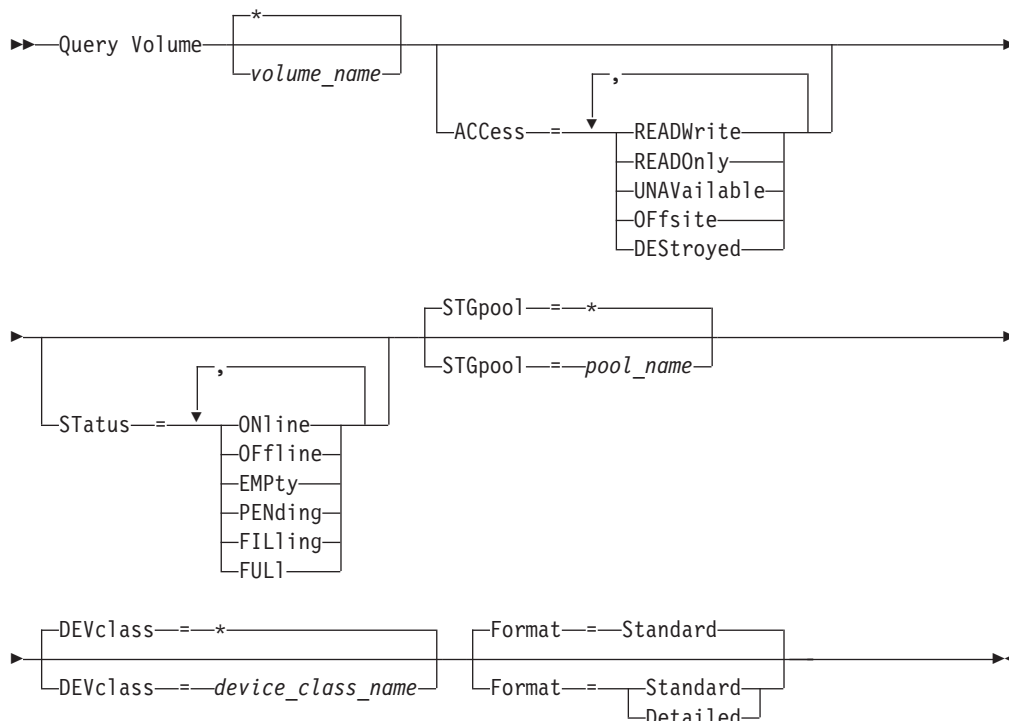
## QUERY VOLUME (Query storage pool volumes)

Use this command to display information about one or more storage pool volumes.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *volume\_name*

Specifies the volume to query. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a name, all storage pool volumes are included in the query.

#### **Access**

Specifies that output is restricted by volume access mode. This parameter is optional. You can specify multiple access modes by separating the modes with commas 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**

Display volumes with an access mode of READWRITE. Client nodes and server processes can read from and write to files stored on the volumes.

#### **READOnly**

Display volumes with an access mode of READONLY. Client nodes and server processes can only read files stored on the volumes.

#### **UNAVailable**

Display volumes with an access mode of UNAVAILABLE. Neither client nodes nor server processes can access files stored on the volumes.



**OFFsite**

Display copy storage pool volumes with an access mode of OFFSITE. The volumes are at offsite locations from which they cannot be mounted.

**DESTROYed**

Display primary storage pool volumes with an access mode of DESTROYED. The volumes have been designated as permanently damaged.

**Status**

Specifies that output is restricted by volume status. This parameter is optional. You can specify multiple status values by separating values with commas 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**

Display random access volumes that are available to the server.

**Offline**

Display random access volumes that are not available to the server.

**EMPTY**

Display sequential access volumes that have no data.

**PENDING**

Display volumes with a status of PENDING. These could be sequential-access volumes from which all files have been deleted, but for which the time specified by the REUSEDELAY parameter on the DEFINE STGPOOL command has not elapsed. These volumes could also be random-access disk volumes that were deleted, but that still contain discarded data that is waiting to be shredded. After the data is shredded, the volume will be physically deleted.

**FILLing**

Display sequential access volumes that the server has written to but has not yet filled to capacity.

**FULL**

Display sequential access volumes that the server has filled.

**STGPool**

Specifies the storage pool to include in the query. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a storage pool name, all storage pools are included in the query.

**DEVclass**

Specifies the device class to include in the query. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a device class name, all devices are included in the query.

**Format**

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.

**Detailed**

Specifies that complete information is displayed.

### Example: List all file storage pool volumes

Display information on all storage pool volumes with the device class name of FILE. See “Field descriptions” on page 821 for field descriptions.

```
query volume devclass=file
```

| Volume Name        | Storage Pool Name | Device Class Name | Estimated Capacity | Pct Util | Volume Status |
|--------------------|-------------------|-------------------|--------------------|----------|---------------|
| /FCT/SERVER/COV011 | COPYSTG           | FILE              | 0.0 M              | 0.0      | Pending       |
| /FCT/SERVER/COV012 | COPYSTG           | FILE              | 0.0 M              | 0.0      | Empty         |
| /FCT/SERVER/COV013 | COPYSTG           | FILE              | 0.0 M              | 0.0      | Empty         |
| /FCT/SERVER/PRV011 | PRIMESTG          | FILE              | 0.0 M              | 0.0      | Empty         |
| /FCT/SERVER/PRV012 | PRIMESTG          | FILE              | 0.0 M              | 0.0      | Empty         |

### Example: Display detailed information about a specific storage pool volume

Display details about the storage pool volume named */fct/server/cov011*. See “Field descriptions” on page 821 for field descriptions.

```
query volume cov011 format=detailed
```

```

      Volume Name: /FCT/SERVER/COV011
      Storage Pool Name: COPYSTG
      Device Class Name: DISK
      Estimated Capacity: 10.0 M
      Scaled Capacity Applied:
      Pct Util: 6.7
      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: 04/14/1998 16:17:26
      Approx. Date Last Read: 04/01/1998 13:26:18
      Date Became Pending:
      Number of Write Errors: 0
      Number of Read Errors: 0
      Volume Location:
      Volume is MVS Lanfree Capable: No
      Last Update by (administrator): COLLIN
      Last Update Date/Time: 05/01/1998 14:07:27
      Begin Reclaim Period:
      End Reclaim Period:
      Drive Encryption Key Manager:

```

### Example: Display detailed information about a storage pool volumes with a specific device class

Display details about a volume in a storage pool with a device class name of FILECLASS. See “Field descriptions” on page 821 for field descriptions.

```
query volume devclass=fileclass format=detailed
```

```

        Volume Name: /WORM_FILESYS/0000000E.BFS
        Storage Pool Name: FILEP00L
        Device Class Name: FILECLASS
        Estimated Capacity: 2.0 G
        Scaled Capacity Applied:
            Pct Util: 0.0
        Volume Status: Filling
            Access: Read/Write
        Pct. Reclaimable Space: 0.0
            Scratch Volume?: Yes
            In Error State?: No
        Number of Writable Sides: 1
        Number of Times Mounted: 1
            Write Pass Number: 1
        Approx. Date Last Written: 03/22/2004 15:23:46
        Approx. Date Last Read: 03/22/2004 15:23:46
            Date Became Pending:
        Number of Write Errors: 0
        Number of Read Errors: 0
            Volume Location:
        Volume is MVS Lanfree Capable: No
        Last Update by (administrator):
            Last Update Date/Time: 03/22/2004 15:23:46
            Begin Reclaim Period: 03/22/2005
            End Reclaim Period: 04/22/2005
        Drive Encryption Key Manager:

```

### Example: Display detailed information about a specific storage pool volume

Display details about a storage pool volume named 000642. The volume is in a storage pool associated with a 3592 device class. See “Field descriptions” for field descriptions.

```
query volume 000642 format=detailed
```

```

        Volume Name: 000642
        Storage Pool Name: 3592P00L
        Device Class Name: 3592CLASS
        Estimated Capacity: 2.0 G
        Scaled Capacity Applied:
            Pct Util: 0.0
        Volume Status: Filling
            Access: Read/Write
        Pct. Reclaimable Space: 0.0
            Scratch Volume?: Yes
            In Error State?: No
        Number of Writable Sides: 1
        Number of Times Mounted: 1
            Write Pass Number: 1
        Approx. Date Last Written: 03/22/2004 15:23:46
        Approx. Date Last Read: 03/22/2004 15:23:46
            Date Became Pending:
        Number of Write Errors: 0
        Number of Read Errors: 0
            Volume Location:
        Volume is MVS Lanfree Capable: No
        Last Update by (administrator):
            Last Update Date/Time: 03/22/2004 15:23:46
            Begin Reclaim Period: 03/22/2005
            End Reclaim Period: 04/22/2005
        Drive Encryption Key Manager: Tivoli Storage Manager

```

### Field descriptions

#### Volume Name

The name of the storage pool volume.

## QUERY VOLUME

### **Storage Pool Name**

The storage pool to which the volume is defined.

### **Device Class Name**

The device class assigned to the storage pool.

### **Estimated Capacity**

The estimated capacity of the volume, in megabytes (M), gigabytes (G), or terabytes (T)

For DISK devices, this value is the capacity of the volume.

For sequential access devices, this value is an estimate of the total space available on the volume, based on the device class.

### **Scaled Capacity Applied**

The percentage of capacity to which a volume is scaled. For example, a value of 20 for a volume whose maximum capacity is 300 GB indicates that the volume can only store 20 percent of 300 GB, or 60 GB. This attribute applies only to IBM 3592 devices.

### **Pct Util**

An estimate of the utilization of the volume. The utilization includes all space occupied by both files and aggregates, including empty space within aggregates.

For DISK volumes, the utilization also includes space occupied by cached data.

### **Volume Status**

The status of the volume.

### **Access**

Whether the volume is available to the server.

### **Pct. Reclaimable Space** (*sequential access volumes only*)

The amount of space on this volume that can be reclaimed because data has expired or been deleted. This value is compared to the reclamation threshold for the storage pool to determine if reclamation is necessary. Reclaimable space includes empty space within aggregates.

For volumes belonging to a SnapLock storage pool, the value is displayed but is not used.

### **Scratch Volume?** (*sequential access volumes only*)

Whether this volume will be returned to scratch when the volume becomes empty.

### **In Error State?**

Whether the volume is in an error state. The server cannot write to volumes in an error state.

### **Number of Writable Sides**

This information is reserved for IBM Tivoli Storage Manager.

### **Number of Times Mounted**

The number of times that the server has opened the volume for use. The number of times that the server has opened the volume is not always the same as the number of times that the volume has been physically mounted in a drive. After a volume is physically mounted, the server can open the same volume multiple times for different operations, for example for different client backup sessions.

**Write Pass Number** (*sequential access volumes only*)

The number of times the volume has been written to from the beginning to the end.

**Approx. Date Last Written**

The approximate date on which the volume was last written.

**Approx. Date Last Read**

The approximate date on which the volume was last read.

**Date Became Pending**

The date that the status of the volume changed to pending.

**Number of Write Errors**

The number of writing errors that have occurred on the volume.

**Number of Read Errors**

The number of reading errors that have occurred on the volume.

**Volume Location**

The location of the volume.

**Volume is MVS Lanfree Capable**

Whether the volume is LAN-free capable. A LAN-free capable volume is one that has been defined and used (at least once) by the Tivoli Storage Manager Version 5 Release 2 or higher z/OS data manager server.

**Last Update by (administrator)**

The administrator that defined or most recently updated the volume.

**Last Update Date/Time**

When the volume was defined or most recently updated.

**Begin Reclaim Period**

Represents the date after which the Tivoli Storage Manager server will begin reclaiming this volume, but not later than the date represented by the end reclaim period. If, when the reclaim period begins, there are files on the volume that have not expired, they will be moved to a new WORM volume during reclamation processing. This field displays a date only if this volume is in a storage pool for which the value of the RECLAMATIONTYPE parameter is SNAPLOCK.

If more than one archive is stored on the same volume, the start of the volume's reclamation period is based on the date of the most recent archive. For SnapLock volumes, the RETVer parameter of the DEFINE COPYGROUP command determines how long an archive is stored. If RETVer is set to 100 days, the volume's reclamation period will start 100 days after the first archive is stored on it. If a second archive is stored on the same volume, the reclamation start date will be adjusted to 100 days after the new archive is stored. If the RETVer value is changed after the first archive is stored, the latest reclamation date will apply for all of the archives on the volume. For example, assume RETVer is set to 100 for an initial archive, but is then changed to 50. If a second archive is stored on the volume three days after the first, the reclamation period will not start until 100 days after the first archive was stored.

**End Reclaim Period**

Represents the date by which the Tivoli Storage Manager must complete reclamation processing on this volume to ensure continued protection of the data. It also represents the Last Access Date physical file attribute in the NetApp Filer, which prevents the file from being deleted until after

## QUERY VOLUME

that date. This field displays a date only if this volume is in a storage pool for which the value of the RECLAMATIONTYPE parameter is SNAPLOCK.

### Drive Encryption Key Manager

The drive encryption key manager. This field applies only to volumes in a storage pool associated with a device type of 3592 or LTO.

### Related commands

*Table 265. Commands related to QUERY VOLUME*

| Command         | Description  |
|-----------------|--|
| DEFINE DEVCLASS | Defines a device class.  |
| DEFINE VOLUME   | Assigns a volume to be used for storage within a specified storage pool. |
| DELETE VOLUME   | Deletes a volume from a storage pool.                                    |
| UPDATE DEVCLASS | Changes the attributes of a device class.                                |
| UPDATE VOLUME   | Updates the attributes of storage pool volumes.                          |
| VARY            | Specifies whether a disk volume is available to the server for use.      |

---

## QUIT (End the interactive mode of the administrative client)

Use this 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, batch, or mount modes of the administrative client.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—QUIT—◄◄

### Parameters

None.

### Example: End an interactive administrative client session

End an administrative client session in the interactive mode.

```
quit
```

### Related commands

None.

## RECLAIM STGPOOL (Reclaim volumes in a sequential-access storage pool)

Use this command to reclaim volumes in a sequential-access storage pool. Reclamation makes the fragmented space on volumes usable again by moving any remaining files from one volume to another volume. This makes the original volume available for reuse. Reclamation will not move inactive versions of backup data from volumes in active-data pools. For storage pools defined with RECLAMATIONTYPE=SNAPLOCK, this command will also delete empty WORM FILE volumes that have exceeded their reclaim period.

For storage pools that use Write Once Read Many (WORM) devices, reclamation is not necessary because WORM volumes are not reusable. However, you can run reclamation to allow the server to consolidate data onto fewer volumes. Volumes emptied by reclamation can be checked out of the library, freeing slots for new volumes.

This command can only be used with sequential access storage pools. The storage pool data format cannot be NETAPPDUMP, CELERRADUMP, or NDMPDUMP. Storage pools defined with a CENTERA device class cannot be reclaimed.

You should only use this command if you are not going to use automatic reclamation for the storage pool. You can set the RECLAIM attribute of the storage pool definition to 100 to prevent automatic reclamation from running. This command will honor the values of the RECLAIMPROCESS, and RECLAIMSTGPOOL attributes of the storage pool definition. This command will also honor the value of the OFFSITERECLAIMLIMIT and RECLAIM attribute of the storage pool definition, if not overridden by the OFFSITERECLAIMLIMIT and THRESHOLD command parameters. Use caution when setting the value of the RECLAIM parameter of an active-data pool to 100, because this will prevent the removal of inactive versions of backup data.

This command creates one or more reclamation processes that can be canceled with the CANCEL PROCESS command. The number of processes is limited by the RECLAIMPROCESS attribute of the storage pool definition. To display information on background processes, use the QUERY PROCESS command.

**Remember:** Issuing this command removes duplicate data in a primary storage pool, copy storage pool, or active-data pool that has been set up for deduplication.

### Privilege class

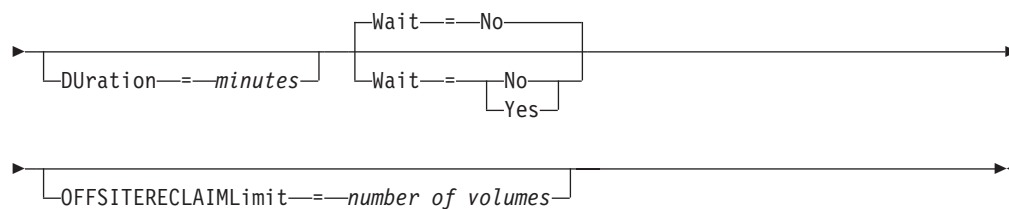
To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool being reclaimed and the reclaim storage pool, if applicable.

### Syntax

```

▶▶ RECLaim STGpool—pool_name—┐
                               └─Threshold=—number—┘
  
```





## Parameters

### *pool\_name* (Required)

Specifies the storage pool in which volumes are to be reclaimed.

### DURATION

Specifies the maximum number of minutes the reclamation will run before being automatically cancelled. You can specify a number from 1 to 9999. This parameter is optional.

After the specified number of minutes elapses, the next time the server checks the reclamation process the server stops the reclamation process. The server checks the reclamation process when the server mounts another eligible volume from the storage pool being reclaimed, and when the server begins to reclaim a new batch of files from the currently mounted volume. As a result, the reclamation can run longer than the value you specified for this parameter.

Until the server checks the reclamation process, there is no indication that the duration period has expired. When the server stops the reclamation process, the server issues message ANR4927W: Reclamation terminated for volume xxx - duration exceeded.

If you do not specify this parameter, the process stops only when no more volumes meet the threshold.

If you specify a duration value for reclamation of a copy storage pool with offsite volumes, you may cause the reclamation to terminate before any volumes are reclaimed. In most situations when initiating reclamation for a copy storage pool with offsite volumes, you should limit the number of offsite volumes to be reclaimed rather than limit the duration. See the OFFSITERECLAIMLIMIT parameter below.

### THRESHOLD

Specifies the percentage of reclaimable space that a volume must have in order to be eligible for reclamation. You can specify a number from 1 to 99. This parameter is optional. If not specified, the RECLAIM attribute of the storage pool definition will be used.

Specify a value of 50 percent or greater for this parameter so files stored on two volumes can be combined into a single output volume.

### OFFSITERECLAIMLIMIT

Specifies the maximum number of offsite storage pool volumes that the server should attempt to reclaim. This parameter is valid only for copy storage pools. Limiting the number of offsite volumes can prevent the server from spending more time analyzing volumes rather than reclaiming them. You can specify a number from 0 to 99999. This parameter is optional. If not specified, the OFFSITERECLAIMLIMIT attribute of the storage pool definition will be used.

## RECLAIM STGPOOL

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is No. Possible values are:

### No

Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed. Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files may have already been moved to new volumes prior to the cancellation.

### Yes

Specifies that the server processes this command in the foreground. The operation must complete before you can continue with other tasks. The server then displays the output messages to the administrative client when the operation completes. Messages are also displayed either in the activity log or the server console, or both, depending on where the messages are logged.

**Note:** You cannot specify WAIT=YES from the server console.

### Example: Reclaim volumes in a sequential-access storage pool

Reclaim volumes in the storage pool named TAPEPOOL. Specify that the server should end the reclamation as soon as possible after 60 minutes.

```
reclaim stgpool tapepool duration=60
```

### Related commands

Table 266. Commands related to RECLAIM STGPOOL

| Command         | Description   |
|-----------------|---|
| CANCEL PROCESS  | Cancels a background server process.                  |
| MIGRATE STGPOOL | Performs migration for the storage pool.              |
| MOVE DRMEDIA    | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA   | Displays information about disaster recovery volumes. |
| QUERY PROCESS   | Displays information about background processes.      |
| QUERY STGPOOL   | QUERY STGPOOL   |

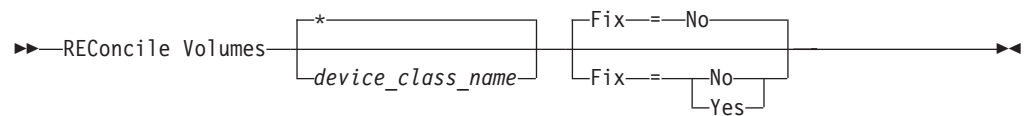
## RECONCILE VOLUMES (Reconcile differences in the virtual volume definitions)

Issue this command from the source server to reconcile differences between virtual volume definitions on the source server and archive files on the target server. Tivoli Storage Manager finds all volumes of the specified device class on the source server and all corresponding archive files on the target server. The target server inventory is also compared to the local definition for virtual volumes to see if inconsistencies exist.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *device\_class\_name*

Specifies the device class name of the virtual volumes. If you do not specify a name, Tivoli Storage Manager reconciles all virtual volumes. This parameter is optional.

#### **FIX**

Specifies whether or not Tivoli Storage Manager attempts to correct any identified inconsistencies. This parameter is optional. The default is NO. Possible values are:

##### **No**

Specifies that Tivoli Storage Manager does not fix any inconsistencies.

##### **Yes**

Specifies that Tivoli Storage Manager makes the following corrections:

- Tivoli Storage Manager marks as unavailable storage pool volumes on the source server that cannot be located on the target server. Volumes that are only found in the volume history, such as database backups and import and export volumes, are reported as being inconsistent.
- Archive files on the target server that do not correspond to any virtual volumes on the source server are marked for deletion from the target server.

## RECONCILE VOLUMES

The following table shows the details of the actions taken:

| FIX= | At the Source Server | At the Target Server                           | Action   |
|------|----------------------|--|--|
| NO   | Volumes exist        | No files exist                                 | Report error   |
|      |                      | Files exist but are marked for deletion        |  |
|      |                      | Active files exist but attributes do not match |  |
|      | Volumes do not exist | Active files exist                             | Report error   |
|      |                      | Files exist but are marked for deletion        | None   |
| YES  | Volumes exist        | No files exist                                 | Report error<br><br><b>Storage pool volumes:</b> Marked as unavailable   |
|      |                      | Files exist but marked for deletion            | Report error<br><br><b>Storage pool volumes:</b> If attributes match, mark files on the target server as active again, mark volumes on the source server as unavailable, and recommend that an AUDIT VOLUME be done to verify the data. If attributes do not match, mark volumes as unavailable. |
|      |                      | Active files exist but attributes do not match | Report error<br><br><b>Storage pool volumes:</b> Mark as unavailable and recommend that an AUDIT VOLUME be done to verify the data.  |
|      | Volumes do not exist | Active files exist                             | Mark files for deletion on the target server.  |
|      |                      | Files exist but marked for deletion            | None   |
|      |                      |  |  |

### Example: Reconcile differences in the virtual volume definitions

Reconcile the differences between all virtual volumes definitions on the source server and archive files on the target server to correct any inconsistencies.

```
reconcile volumes remotel fix=yes
```

### Related commands

Table 267. Commands related to RECONCILE VOLUMES

| Command         | Description   |
|-----------------|---|
| DEFINE DEVCLASS | Defines a device class.                               |
| DEFINE SERVER   | Defines a server for server-to-server communications. |
| DELETE SERVER   | Deletes the definition of a server.                   |
| QUERY SERVER    | Displays information about servers.                   |

*Table 267. Commands related to RECONCILE VOLUMES (continued)*

| Command       | Description                         |
|---------------|-------------------------------------|
| UPDATE SERVER | Updates information about a server. |

### REGISTER commands

Use the REGISTER commands to define or add objects to Tivoli Storage Manager.

The following is a list of REGISTER commands for Tivoli Storage Manager:

- “REGISTER ADMIN (Register an administrator)” on page 833
- “REGISTER LICENSE (Register a new license)” on page 835
- “REGISTER NODE (Register a node)” on page 837



**Yes**

Specifies that the administrator's password will expire at the next sign-on. The client or administrator must change or reset the password at that time. If a password is not specified, you will receive an error message.

**EMAILADDRESS**

This parameter is used by IBM Tivoli Storage Manager Express only. It is not currently used by Tivoli Storage Manager. Tivoli Storage Manager Express uses this parameter to specify the e-mail address or addresses responsible for this administrator. Any reports that are directed to this administrator are sent to these e-mail addresses.

**Example: Register an administrator**

Define an administrator, LARRY, with the password PASSONE. You can identify LARRY as second-shift personnel by specifying this information with the CONTACT parameter. Issue the command:

```
register admin larry passone contact='second shift'
```

**Related commands**

*Table 268. Commands related to REGISTER ADMIN*

| Command         | Description   |
|-----------------|---|
| GRANT AUTHORITY | Assigns privilege classes to an administrator.                                      |
| LOCK ADMIN      | Prevents an administrator from accessing IBM Tivoli Storage Manager.                |
| QUERY ADMIN     | Displays information about one or more IBM Tivoli Storage Manager administrators.   |
| REGISTER NODE   | Defines a client to the server and sets options for that user.                      |
| REMOVE ADMIN    | Removes an administrator from the list of registered administrators.                |
| RENAME ADMIN    | Changes an IBM Tivoli Storage Manager administrator's name.                         |
| SET PASSEXP     | Specifies the number of days after which a password is expired and must be changed. |
| UNLOCK ADMIN    | Enables a locked administrator to access IBM Tivoli Storage Manager.                |
| UPDATE ADMIN    | Changes the password or contact information associated with any administrator.      |
| UPDATE NODE     | Changes the attributes associated with a client node.                               |



## REGISTER LICENSE (Register a new license)

Use this command to register new licenses for server components, including Tivoli Storage Manager (base), Tivoli Storage Manager Extended Edition, and System Storage Archive Manager.

Licenses are stored in enrollment certificate files. The enrollment certificate files contain licensing information for the server product. The nodelock file preserves the licensing information for your installation. Your license agreement determines what you are licensed to use, even if you cannot use the REGISTER LICENSE command to register all components. You are expected to comply with the license agreement and use only what you have purchased. Use of the REGISTER LICENSE command implies that you agree to and accept the license terms specified in your license agreement.

### Important:

- Before upgrading from a previous version of Tivoli Storage Manager, you must delete or rename the nodelock file.
- To unregister licenses, you must erase the nodelock file found in the server directory of your installation. This will also require you to reregister any previously registered licenses.
- You cannot register licenses for components that are licensed on the basis of processors (for example, Tivoli Storage Manager for Mail, Tivoli Storage Manager for Databases, Tivoli Storage Manager for ERP, and Tivoli Storage Manager for Space Management).

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶—REGister LICense—FILE—=—
                                |
                                |—tsmbasic.lic
                                |—tsmee.lic
                                |—dataret.lic
                                |—*.lic
                                |
                                |—————▶▶
  
```

### Parameters

#### FILE

Specifies the name of the enrollment certificate file containing the license to be registered. The specification can contain a wildcard (\*). Enter the complete file name or a wildcard in place of the file name. The file names are case-sensitive. Possible values are:

#### tsmbasic.lic

To license base IBM Tivoli Storage Manager.

#### tsmee.lic

To license IBM Tivoli Storage Manager Extended Edition. This includes the disaster recovery manager, large libraries, and NDMP.

#### dataret.lic

To license System Storage Archive Manager. This is required to enable Data Retention Protection as well as Expiration and Deletion Suspension (Deletion Hold).

## REGISTER LICENSE

**\*.lic**

To license all IBM Tivoli Storage Manager licenses for server components.

### Example: Register a license

Register System Storage Archive Manager license.

```
register license file=dataret.lic
```

### Example: Register all licenses

Register all license files using a wildcard.

```
register license file=*.lic
```

### Related commands

*Table 269. Commands related to REGISTER LICENSE*

| Command                | Description   |
|------------------------|---|
| AUDIT LICENSES         | Checks for compliance with defined licenses.  |
| QUERY LICENSE          | Displays information about licenses and audits.   |
| QUERY STATUS           | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET LICENSEAUDITPERIOD | Specifies the number of days between automatic license audits.                          |

## REGISTER NODE (Register a node)

Use this command to register a node to the server.

This command also automatically creates an administrative user ID with client owner authority over the node. You can use this administrative user ID to access the Web backup-archive client from remote locations through a Web browser. If an administrative user ID already exists with the same name as the node being registered, an administrative user ID is not automatically defined. The client node is registered without an administrative user ID. This process also applies if your site uses open registration.

If a client requires a different policy domain than STANDARD, you must register the client node with this command or update the registered node.

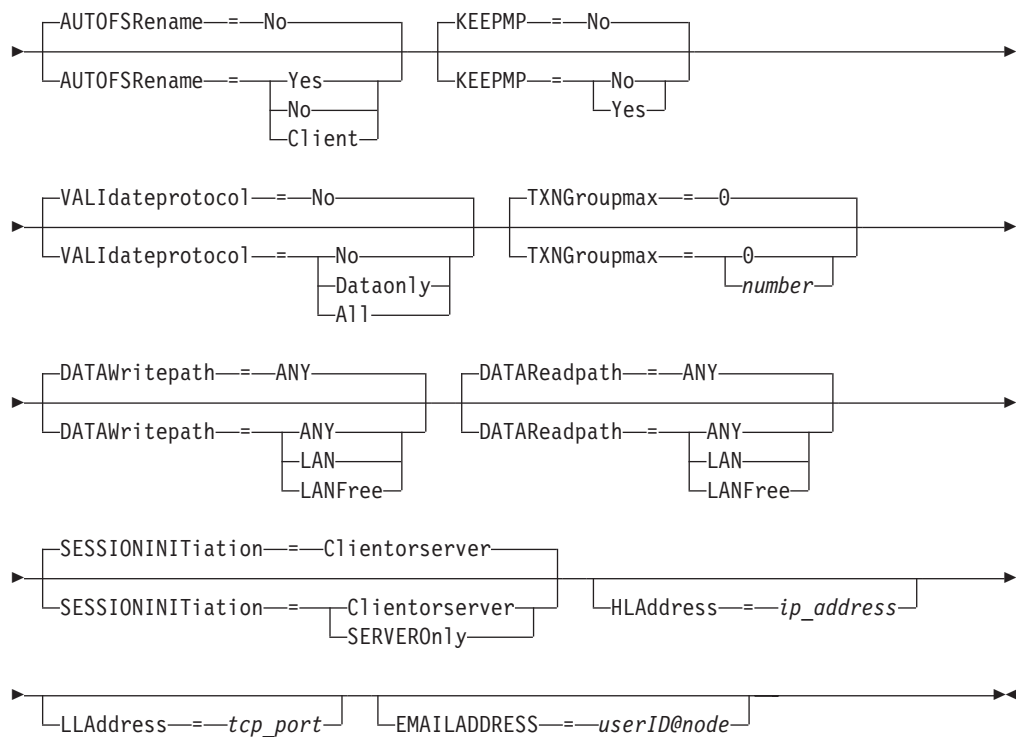
### Privilege class

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 is assigned.

### Syntax



## REGISTER NODE



### Parameters

#### *node\_name* (Required)

Specifies the name of the client node to be registered. The maximum length of the name is 64 characters.

You cannot specify a node name of NONE.

#### *password* (Required)

Specifies the client node's password. The maximum length of the name is 64 characters. The password is not case-sensitive. See "Naming Tivoli Storage Manager objects" on page 12 for a table that lists the characters available for specifying a password.

#### PASSExp

Specifies the number of days the password remains valid. You can set the password expiration period from 0 to 9999 days. A value of 0 means that the password never expires. This parameter is optional. If you do not specify this parameter, the server's common password expiration period is used. The common password expiration period is 90 days unless changed using the SET PASSEXP command.

You can change the password expiration period by using the UPDATE NODE or SET PASSEXP commands. The SET PASSEXP command enables you to set a common expiration period for all administrators and client nodes, or you may use it to selectively set password expiration periods. If you selectively set a password expiration period by using the REGISTER NODE command, the UPDATE NODE command, or the SET PASSEXP command, the expiration period is excluded from common password expiration periods that were created using the SET PASSEXP command.

You can use the RESET PASSEXP command to reset the password expiration period to the common expiration period.

**USerid**

Specifies the administrative user ID with client owner authority. If you do not specify a user ID, by default an administrative user ID with client owner authority is created using the nodename of the client node to be registered. This parameter is optional. When PASSWORDACCESS=GENERATE is used by the client to change the password, the administrative ID with the same name can be used to access the Web backup-archive client from a remote location. Possible values are:

**NONE**

Specifies that an administrative user ID will not be automatically defined.

*user\_id*

Specifies a node ID different than the node being registered. You can use this parameter to grant client owner authority to an existing administrative user ID.

**CONtact**

Specifies a text string of information identifying the node. The 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**

Specifies the name of the policy domain to which the node is assigned. The parameter is optional. If you do not specify a policy domain name, the node is assigned to the default policy domain (STANDARD).

When a source server is registered as a node, it is assigned to a policy domain. Data from the source server is stored in the storage pool specified in the archive copy group of the default management class of that domain.

**COMPression**

Specifies whether the client node compresses its files before sending them to the server for backup and archive. The parameter is optional. The default value is CLIENT.

**Note:** This parameter cannot be specified for a NAS node.

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**

Specifies whether the client node can delete its own archive files from the server. The 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.

## REGISTER NODE

### **BACKDELeTe**

Specifies whether the client node can delete its own backup files from the server. The 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.

### **CLOptset**

Specifies the name of the option set to be used by the client. The parameter is optional.

### **FORCEPwreset**

Specifies whether to force a client to change or reset the password. The parameter is optional. The default value is NO. Possible values are:

#### **No**

Specifies that the password expiration period is set by the SET PASSEXP command. The client does not need to change or reset the password while attempting to logon to the server.

#### **Yes**

Specifies that the client node password will expire at the next logon. The client must change or reset the password at that time. If a password is not specified, you will receive an error message.

### **Type**

Specifies the type of node being registered. The parameter is optional. The default value is CLIENT. Possible values are:

#### **Client**

Specifies that the client node is a backup-archive client, Tivoli Storage Manager for Space Management client, or application client.

#### **NAS**

Specifies that the node is a network-attached storage (NAS) file server whose data will be protected using NDMP operations. The node name cannot be SERVER.

**Note:** The name of the NAS node must be the same as the data mover. Therefore, the name cannot be changed after a corresponding data mover is defined.

#### **Server**

Specifies that the client node is a source server being registered on the target server.

### **URL**

Specifies the URL address that is used in your Web browser to administer the client node. The parameter is optional.

### **MAXNUMMP**

Specifies the maximum number of mount points a node is allowed to use on the server or storage agent only for operations such as backup, archive, and Tivoli Storage Manager for Space Management migration. The parameter is optional and does not apply to nodes with a type of NAS or SERVER. The default value is 1. You can specify an integer from 0–999. A value of 0 specifies that a node cannot acquire any mount point for a client data store operation.

The MAXNUMMP value is not evaluated or enforced during client data read operations such as restore, retrieve, and Tivoli Storage Manager for Space Management recall. However, mount points in use for data read operations are evaluated against attempted concurrent data store operations for the same client node and may prevent the data store operations from being able to acquire mount points.

For volumes in a storage pool associated with a FILE device type, the server can have multiple processes to read and one process to write to the same volume concurrently. To provide efficient access for nodes with data in FILE-type storage pools, increase the value of the MAXNUMMP parameter to create greater concurrency when accessing FILE volumes.

For nodes that store data into primary storage pools with simultaneous write enabled, you may need to adjust the value of the MAXNUMMP parameter to specify the correct number of mount points for each client session. For details, refer to information about simultaneous write in the *Administrator's Guide*.

For server-to-server backup, if one server is at a different version than the other server, set the number of mount points on the target server to a value higher than 1. Otherwise, you will receive an error.

A storage agent independently tracks the number of points used during a client session. If a node has a storage agent installed it may exceed the MAXNUMMP value. The MAXNUMMP value may also be exceeded under conditions where the node does not have to wait for a mount point.

**Note:** The server may preempt a client's operation for a higher priority operation and the client may lose a mount point if no other mount points are available. Refer to the *Administrator's Guide* for more information.

### KEEPMP

Specifies whether the client node keeps the mount point for the entire session. The parameter is optional. The default value is NO. Possible values are:

#### Yes

Specifies that the client node must retain the mount point during the entire session. If policy definitions cause data to be stored to a disk storage pool after storing data to a sequential access storage pool, any mount points held by the session will not be released.

#### No

Specifies that the client node release the mount point during the session. If policy definitions cause data to be stored to a disk storage pool after storing data to a sequential access storage pool, any mount points held by the session will be released.

### AUTOFSRename

Specify whether file spaces are automatically renamed when you upgrade the client system to support Unicode or specify if file spaces are renamed by the client, if needed. The default is NO. Setting the parameter to YES enables automatic renaming, which occurs when the client runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup. The automatic renaming changes the names of existing backed-up file spaces that are not in Unicode in server storage. Then the file spaces are backed up in Unicode. You can use this parameter for Unicode-enabled Tivoli Storage Manager clients using Windows, Macintosh OS X, and NetWare operating systems.

After the client with support for Unicode is installed, any new file spaces that the client backs up are stored in server storage using the UTF-8 code page. UTF-8 is a byte-oriented encoding form specified by the Unicode Standard.

Possible values are:

### Yes

Existing file spaces are automatically renamed when you upgrade to a client that supports Unicode and the client runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup. The renaming occurs whether the client uses the graphical user interface, the command line, or the client scheduler.

For example, the server renames a drive as follows:

Original name: D\_DRIVE  
New name: D\_DRIVE\_OLD

The new name indicates that the file space is stored on the server in a format that is not Unicode.

### No

Existing file spaces are not automatically renamed when the client system upgrades to a client that supports Unicode, and the client runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup.

### Client

The option AUTOFSRENAME in the client's option file determines whether file spaces are renamed.

By default, the client option is set to PROMPT. When the client system upgrades to a client that supports Unicode and the client runs a Tivoli Storage Manager operation with the graphical user interface or the command line, the program displays a one-time prompt to the user about whether to rename file spaces.

When the client scheduler runs an operation, the program does not prompt for a choice about renaming, and does not rename file spaces. Backups of existing file spaces are sent as before (not in Unicode).

Refer to the appropriate *Backup-Archive Clients Installation and User's Guide* book for more information on the client option.

## VALIDATEprotocol

Specify whether Tivoli Storage Manager performs a cyclic redundancy check (CRC) to validate the data sent between the client and server. The parameter is optional. The default is NO. Possible values are:

### No

Specifies that data validation not be performed on any data sent between the client and server.

### Dataonly

Specifies that data validation be performed only on file data that is sent between the client and server. This does not include the file metadata. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the client and the server.

### All

Specifies that data validation be performed on all client file data, client file metadata, and Tivoli Storage Manager server metadata that is sent between



the client and server. This mode impacts performance as additional overhead is required to calculate and compare CRC values between the client and the server.

### TXNGroupmax

Specifies the number of files per transaction commit that are transferred between a client and a server. Client performance may be improved by using a larger value for this option.

The default value is 0. Specifying 0 indicates the node will use the server global value that is set in the server options file. To use a value other than the server global value, specify a value of 4 through 65,000 for this parameter. The node value takes precedence over the server value.

**Attention:** Increasing the TXNGROUPMAX value will result in increased recovery log utilization. Higher recovery log utilization may increase the risk of running out of log space. Evaluate each node's performance before changing this parameter. For more information on managing the recovery log, see the *Administrator's Guide*.

### DATAWritepath

Specifies the transfer path used when the client sends data to the server, storage agent, or both, during storage operations such as backup or archive. The parameter is optional. The default is ANY.

**Note:** If a path is unavailable, the node cannot send any data. For example, if you select the LAN-free option but a LAN-free path is not defined, the operation will not work.

Possible values are:

#### ANY

Specifies that data is sent to the server, storage agent, or both, by any available path. A LAN-free path will be used if one is available. If a LAN-free path is unavailable, the data will be moved using the LAN.

#### LAN

Specifies that data is sent using the LAN.

#### LANFree

Specifies that data is sent using a LAN-free path.

### DATAReadpath

Specifies the transfer path used when the server, storage agent, or both read data for a client, during operations such as restore or retrieve. The parameter is optional. The default is ANY.

**Note:** If a path is unavailable, data cannot be read. For example, if you select the LAN-free option but a LAN-free path is not defined, the operation will not work.

Possible values are:

#### ANY

Specifies that the server, storage agent, or both use any available path to read data. A LAN-free path will be used if one is available. If a LAN-free path is unavailable, the data will be read using the LAN.

#### LAN

Specifies that data is read using the LAN.

## REGISTER NODE

### **LANFree**

Specifies that data is read using a LAN-free path.

### **SESSIONINITiation**

Controls whether the server or the client initiates sessions. The default is that the client initiates sessions. The parameter is optional.

### **Clientorserver**

Specifies that the client may initiate sessions with the server by communicating on the TCP/IP port defined with the server option TCPPORT. Server-prompted scheduling may also be used to prompt the client to connect to the server.

### **SERVEROnly**

Specifies that the server will not accept client requests for sessions. All sessions must be initiated by server-prompted scheduling on the port defined for the client with the REGISTER or UPDATE NODE commands. You cannot use the client acceptor (dsmcad) to start the scheduler when SESSIONINITIATION is set to SERVERONLY.

### **HLAddress**

Specifies the client IP address that the server contacts to initiate scheduled events. This optional parameter is used only when SESSIONINITIATION is set to SERVERONLY, regardless of any addresses previously used by the client to contact the server. If SESSIONINITIATION SERVERONLY is not in use, this option has no effect.

The address can be specified either in numeric or host name format. If a numeric address is used, it will be saved without verification by a domain name server. If the address is not correct, it can cause failures when the server attempts to contact the client. Host name format addresses will be verified with a domain name server. Verified names will be saved and resolved with Domain Name Services when the server contacts the client.

### **LLAddress**

Specifies the client port number on which the client listens for sessions from the server. This optional parameter is used only when SESSIONINITIATION is set to SERVERONLY, regardless of any addresses previously used by the client to contact the server. If SESSIONINITIATION SERVERONLY is not in use, this option has no effect.

The value for this parameter must match the value of client option TCPCLIENTPORT. The default value is 1501.

### **EMAILADDRESS**

This parameter is used by IBM Tivoli Storage Manager Express only. It is not currently used by Tivoli Storage Manager. Tivoli Storage Manager Express uses this parameter to specify the e-mail address or addresses for this administrator. Any reports that are directed to this administrator are sent to these e-mail addresses.

### **Example: Register a client node and password and set compression on**

Register the client node J0E0S2 with the password *SECRETCODE* and assign this node to the DOM1 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. This command automatically creates a J0E0S2 administrative

user ID with password *SECRETCODE*. In addition, the administrator now has client owner authority to the JOEOS2 node.

```
register node joeos2 secretcode domain=dom1
archdelete=yes backdelete=yes
compression=yes
```

### Example: Register a client node and password and prevent the administrative ID from being automatically assigned

Register the client node FRAN and prevent an administrative user ID from being automatically defined

```
register node fran xy34z userid=none
```

### Example: Grant client owner authority for an existing administrative user

Grant client owner authority to an existing administrative user ID, *HELPADMIN*, when registering the client node JAN. This would not automatically create an administrator ID named JAN, but would grant client owner authority for this node to the *HELPADMIN* administrator.

```
register node jan pwdsafe userid=helpadmin
```

### Example: Register a NAS file server node that uses NDMP operations

Register a node name of NAS1 for a NAS file server that is using NDMP operations. Assign this node to a special NAS domain.

```
register node nas1 pw4pw domain=nasdom type=nas
```

### Example: Register a node and specify data validation

Register a node name of ED whose data is backed up to a primary storage pool that also has a list of 2 copy storage pools defined. The primary storage pool is of disk device type, and the 2 copy storage pools are of sequential device type. Specify 2 as the maximum number of mount points. The client's network has been unstable during the last few weeks. The client wants assurance that his data is not corrupted as it is sent over the network. Specify that Tivoli Storage Manager should validate all data that is sent by this user to the server.

```
register node ed pw45twx maxnummp=2 validateprotocol=all
```

### Example: Register a node and specify the maximum number of files per transaction commit

Register a node name of ED and set the TXNGroupmax to 1,000.

```
register node ed pw45twx txngroupmax=1000
```

## Related commands

Table 270. Commands related to REGISTER NODE

| Command            | Description  |
|--------------------|--|
| DEFINE ASSOCIATION | Associates clients with a schedule.                            |
| DEFINE DATAMOVER   | Defines a data mover to the IBM Tivoli Storage Manager server. |

## REGISTER NODE

Table 270. Commands related to REGISTER NODE (continued)

| Command                    | Description  |
|----------------------------|--|
| DEFINE MACHNODEASSOCIATION | Associates an IBM Tivoli Storage Manager node with a machine.                              |
| DELETE FILESPACE           | Deletes data associated with client's file spaces.   |
| LOCK NODE                  | Prevents a client from accessing the server.   |
| REGISTER ADMIN             | Defines a new administrator without granting administrative authority.                     |
| REMOVE NODE                | Removes a client from the list of registered nodes for a specific policy domain.           |
| RENAME NODE                | Changes the name for a client node.  |
| RESET PASSEXP              | Resets the password expiration for nodes or administrators.                                |
| QUERY NODE                 | Displays partial or complete information about one or more clients.                        |
| SET PASSEXP                | Specifies the number of days after which a password is expired and must be changed.        |
| SET REGISTRATION           | Specifies whether users can register themselves or must be registered by an administrator. |
| UNLOCK NODE                | Enables a locked user in a specific policy domain to access the server.                    |
| UPDATE ADMIN               | Changes the password or contact information associated with any administrator.             |
| UPDATE NODE                | Changes the attributes associated with a client node.                                      |

---

## REMOVE commands

Use the REMOVE commands to remove an object from Tivoli Storage Manager.

The following is a list of REMOVE commands for Tivoli Storage Manager:

- “REMOVE ADMIN (Delete an administrator)” on page 848
- “REMOVE NODE (Delete a node or an associated machine node)” on page 849

### REMOVE ADMIN (Delete an administrator)

Use this command to remove an administrator from the system.

You cannot remove the last system administrator or the SERVER\_CONSOLE administrative ID from the system.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

►►—REMOve Admin—*admin\_name*—◄◄

#### Parameters

*admin\_name* **(Required)**

Specifies the name of the administrator to be removed.

#### Example: Remove an administrator

Remove the administrator LARRY. Issue the following command:

```
remove admin larry
```

#### Related commands

Table 271. Commands related to REMOVE ADMIN

| Command        | Description   |
|----------------|---|
| LOCK ADMIN     | Prevents an administrator from accessing IBM Tivoli Storage Manager.              |
| QUERY ADMIN    | Displays information about one or more IBM Tivoli Storage Manager administrators. |
| REGISTER ADMIN | Defines a new administrator without granting administrative authority.            |
| RENAME ADMIN   | Changes an IBM Tivoli Storage Manager administrator's name.                       |

## REMOVE NODE (Delete a node or an associated machine node)

Use this command to remove a node from the server. If you are using disaster recovery manager and the node to be removed is associated with a machine, the association between the node and the machine is also deleted. If a node is a member of a collocation group and you remove the node from the server, the node is also removed from the collocation group.

When a node is removed, the corresponding administrative ID is removed only if:

- The administrator name is identical to the node name.
- The administrator has client owner or client access authority *only* to the node being removed.
- The administrator is not a managed object.

**Important:** Before you can remove a node, you must delete all backup and archive file spaces that belong to that node.

Before you can remove a NAS node that has a corresponding data mover defined, you must first delete any paths from the data mover, delete the data mover, delete all virtual file space definitions for the node, then remove the NAS 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 is assigned.

### Syntax

►►—REMove Node—*node\_name*—————►►

### Parameters

#### *node\_name* (Required)

Specifies the name of the node to be removed.

### Example: Remove a client node

Remove the client node LARRY.

```
remove node larry
```

### Related commands

Table 272. Commands related to REMOVE NODE

| Command                    | Description  |
|----------------------------|--|
| DELETE MACHNODEASSOCIATION | Deletes association between a machine and node.    |
| DELETE DATAMOVER           | Deletes a data mover.                              |
| DELETE FILESPACE           | Deletes data associated with client's file spaces. |
| DELETE PATH                | Deletes a path from a source to a destination.     |
| DELETE VIRTUALFSMAPPING    | Delete a virtual file space mapping.               |

## REMOVE NODE

*Table 272. Commands related to REMOVE NODE (continued)*

| Command           | Description  |
|-------------------|--|
| LOCK NODE         | Prevents a client from accessing the server.   |
| QUERY COLLOCGROUP | Displays information about collocation groups.   |
| QUERY MACHINE     | Displays information about machines.   |
| QUERY NODE        | Displays partial or complete information about one or more clients.                                      |
| QUERY SESSION     | Displays information about all active administrator and client sessions with IBM Tivoli Storage Manager. |
| REGISTER NODE     | Defines a client to the server and sets options for that user.   |
| RENAME NODE       | Changes the name for a client node.  |



## RENAME commands

---

Use the RENAME commands to change the name of an existing object.

The following is a list of RENAME commands for Tivoli Storage Manager:

- “RENAME ADMIN (Rename an administrator)” on page 852
- “RENAME FILESPACE (Rename a client file space on the server)” on page 853
- “RENAME NODE (Rename a node)” on page 856
- “RENAME SCRIPT (Rename a Tivoli Storage Manager script)” on page 857
- “RENAME SERVERGROUP (Rename a server group)” on page 858
- “RENAME STGPOOL (Change the name of a storage pool)” on page 859

### RENAME ADMIN (Rename an administrator)

Use this 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.

You cannot rename the SERVER\_CONSOLE administrative ID.

#### Privilege class

To issue this command, you must have system privilege.

#### Syntax

►►—REName Admin—*current\_admin\_name*—*new\_admin\_name*—◄◄

#### Parameters

*current\_admin\_name* **(Required)**

Specifies the name of the administrator to be renamed.

*new\_admin\_name* **(Required)**

Specifies the new administrator name. The maximum length of the name is 64 characters.

#### Example: Rename an administrator

Rename the Tivoli Storage Manager administrator CLAUDIA to BILL.

```
rename admin claudia bill
```

#### Related commands

Table 273. Commands related to RENAME ADMIN

| Command      | Description   |
|--------------|---|
| QUERY ADMIN  | Displays information about one or more IBM Tivoli Storage Manager administrators. |
| UPDATE ADMIN | Changes the password or contact information associated with any administrator.    |

# RENAME FILESPACE (Rename a client file space on the server)

Use this command to rename an existing client file space on the server to a new file space name or to rename imported file spaces.

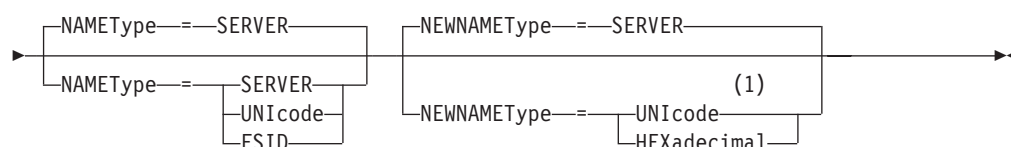
You might want to rename a file space that has been imported or to cause the creation of new Unicode-enabled file spaces for Unicode-enabled clients.

## 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—*node\_name*—*current\_file\_space\_name*—*new\_file\_space\_name*—►►



## Notes:

- 1 This parameter is the default when you specify NAMETYPE=UNICODE.

## Parameters

### *node\_name* (Required)

Specifies the name of the client node to which the file space to be renamed belongs.

### *current\_file\_space\_name* (Required)

Specifies the name of the file space to be renamed. A file space name is case-sensitive and must be specified exactly as defined to the server. Virtual file space mapping names are allowed.

### *new\_file\_space\_name* (Required)

Specifies the new name for the file space. A client file space name is case-sensitive and must be specified exactly as it is to be defined to the server. This parameter cannot be an existing virtual file space mapping name. If the *current\_file\_space\_name* is a virtual file space, the *new\_file\_space\_name* must follow all the rules for defining a virtual file space name. See the DEFINE VIRTUALFSMAPPING command for more information.

**Attention:** If the new name type is hexadecimal, be careful to specify valid UTF-8 hexadecimal values so the server's code page displays the file space name as intended. For example, do not specify a value that can be interpreted as a backspace.

## NAMETYPE

Specify how you want the server to interpret the current file space name that you enter. This parameter is useful when the server has clients with support for Unicode. You can use this parameter for Unicode-enabled Tivoli Storage Manager clients using Windows, Macintosh OS X and NetWare operating systems.

## RENAME FILESPACE

The default value is SERVER. If a virtual file space mapping name is specified, you must use SERVER. Possible values are:

### SERVER

The server uses the server's code page to interpret the file space name.

### UNICODE

The server converts the file space name entered from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the name and the server's code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has a problem accessing system conversion routines.

### FSID

The server interprets the file space name as the file space ID (FSID).

### NEWNAMETYPE

Specify how you want the server to interpret the new file space name that you enter. The default is SERVER if you specified the NAMETYPE as SERVER, or if the file space to be renamed is not Unicode. The default is UNICODE if you specified the NAMETYPE as UNICODE, or if the file space to be renamed is Unicode. If a virtual file space mapping name is specified, you must use SERVER. Possible values are:

### SERVER

The server uses the server's code page to interpret the file space name.

### UNICODE

The server converts the file space name entered from the server code page to the UTF-8 code page. The success of the conversion depends on the actual characters in the name and the server's code page. If the conversion is not successful, you may want to specify the HEXADECIMAL parameter.

### HEXadecimal

The server interprets the file space name that you enter as the hexadecimal representation of a name in Unicode. Using hexadecimal ensures that the server is able to correctly rename the file space, regardless of the server's code page.

To view the hexadecimal representation of a file space name, you can use the QUERY FILESPACE command with FORMAT=DETAILED.

**Restriction:** You cannot specify a new name of a type that is different from the original name. You can rename a file space that is Unicode to another name in Unicode. You can rename a file space that is not Unicode using a new name in the server's code page. You cannot mix the two types.

### Example: Rename an imported file space to prevent overwriting

An AIX client node named LARRY backed up file space /r033 to the Tivoli Storage Manager server. The file space was exported to tape and later re-imported to the server. When this file space was imported, a system-generated name, /r031, was created for it because /r033 already existed for client node LARRY.

Client node LARRY, however, already had a file space named /r031 that was not backed up, therefore, was unknown to the server. Unless the imported file space is renamed, it will overlay file space /r031 because the file space name generated by the IMPORT function is the same as a file space on client node LARRY that is unknown to the server.

Use the following command to rename imported file space /r031. The new name, /imported-r033, identifies that the new file space is an imported image of file space /r033.

```
rename filespace larry /r031 /imported-r033
```

### Example: Rename file space to create a Unicode-enabled file space

Client JOE is using an English Unicode-enabled Tivoli Storage Manager client. JOE has backed up several large file spaces that are not Unicode enabled in server storage. File space \\joe\c\$ contains some files with Japanese file names that cannot be backed up to a file space that is not Unicode enabled. Because the file spaces are large, the administrator does not want to convert all of JOE's file spaces to Unicode-enabled file spaces at this time. The administrator wants to only rename the non-Unicode file space, \\joe\c\$, so that the next backup of the file space causes the creation of a new Unicode-enabled file space. The new Unicode-enabled file space will allow the Japanese files to be successfully backed up.

Use the following command to rename \\joe\c\$:

```
rename filespace joe \\joe\c$ \\joe\c$_old
```

### Related commands

*Table 274. Commands related to RENAME FILESPACE*

| Command                 | Description   |
|-------------------------|---|
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping.                                    |
| DELETE FILESPACE        | Deletes data associated with client's file spaces.                      |
| EXPORT NODE             | Copies client node information to external media.                       |
| QUERY FILESPACE         | Displays information about data in file spaces that belong to a client. |
| QUERY OCCUPANCY         | Displays file space information by storage pool.                        |

## RENAME NODE (Rename a node)

Use this command to rename a node.

If you are assigning an existing node ID to another person, it is recommended that you use the UPDATE NODE command to change the password.

**Important:** You cannot rename a NAS node name that has a corresponding data mover defined. If the data mover has defined paths, the paths must first be deleted. For details on the steps that are required when you want to rename a NAS node, refer to the *Administrator's Guide*.

### 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—*current\_node\_name*—*new\_node\_name*—►►

### Parameters

*current\_node\_name* **(Required)**

Specifies the name of the node to be renamed.

*new\_node\_name* **(Required)**

Specifies the new name of the node. The maximum length of the name is 64 characters.

### Example: Rename a node

Rename the node JOE to JOYCE.

```
rename node joe joyce
```

### Related commands

Table 275. Commands related to RENAME NODE

| Command     | Description   |
|-------------|---|
| QUERY NODE  | Displays partial or complete information about one or more clients. |
| UPDATE NODE | Changes the attributes associated with a client node.               |

# RENAME SCRIPT (Rename a Tivoli Storage Manager script)

Use this command to rename a Tivoli Storage Manager script.

## Privilege class

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

## Syntax

►►—REName SCRIpt—*current\_script\_name*—*new\_script\_name*—◄◄

## Parameters

*current\_script\_name* **(Required)**

Specifies the name of the script to rename.

*new\_script\_name* **(Required)**

Specifies the new name for the script. The name can contain as many as 30 characters.

## Example: Rename a script

Rename SCRIPT1 to a new script named SCRIPT2.

```
rename script script1 script2
```

## Related commands

Table 276. Commands related to RENAME SCRIPT

| Command       | Description  |
|---------------|--|
| COPY SCRIPT   | Creates a copy of a script.                                |
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| DELETE SCRIPT | Deletes the script or individual lines from the script.    |
| QUERY SCRIPT  | Displays information about scripts.                        |
| RUN           | Runs a script.   |
| UPDATE SCRIPT | Changes or adds lines to a script.                         |

## RENAME SERVERGROUP (Rename a server group)

Use this command to rename a server group.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—RENAME SERVERGRoup—*current\_group\_name*—*new\_group\_name*—◄◄

### Parameters

*current\_group\_name* **(Required)**

Specifies the server group to rename.

*new\_group\_name* **(Required)**

Specifies the new name of the server group. The maximum length of the name is 64 characters.

### Example: Rename a server group

Rename server group WEST\_COMPLEX to BIG\_WEST.

```
rename servergroup west_complex big_west
```

### Related commands

Table 277. Commands related to RENAME SERVERGROUP

| Command            | Description                               |
|--------------------|---|
| COPY SERVERGROUP   | Creates a copy of a server group.         |
| DEFINE SERVERGROUP | Defines a new server group.               |
| DELETE SERVERGROUP | Deletes a server group.                   |
| QUERY SERVERGROUP  | Displays information about server groups. |
| UPDATE SERVERGROUP | Updates a server group.                   |



## RENAME STGPOOL (Change the name of a storage pool)

Use this command to change the name of a storage pool. You can change storage pool names so that the same names can be used on a configuration manager and its managed servers.

When you change a storage pool name, any administrators with restricted storage privilege for the old storage pool are automatically given restricted storage privilege for the renamed storage pool. However, if a management class or copy group specifies the existing storage pool as the destination for files, the destination is not changed to the new storage pool name.

If processes are active when a storage pool is renamed, the old name may still be displayed in messages or queries for those processes.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—REName STGpool—*current\_pool\_name*—*new\_pool\_name*—►►

### Parameters

*current\_pool\_name* **(Required)**

Specifies the storage pool to rename.

*new\_pool\_name* **(Required)**

Specifies the new name of the storage pool. The maximum length of the name is 30 characters.

### Example: Change the name of a storage pool

Rename storage pool STGPOOLA to STGPOOLB:

```
rename stgpool stgpoola stgpoolb
```

### Related commands

Table 278. Commands related to RENAME STGPOOL

| Command         | Description   |
|-----------------|---|
| BACKUP STGPOOL  | Backs up a primary storage pool to a copy storage pool.               |
| DEFINE STGPOOL  | Defines a storage pool as a named collection of server storage media. |
| DELETE STGPOOL  | Deletes a storage pool from server storage.                           |
| QUERY STGPOOL   | Displays information about storage pools.                             |
| RESTORE STGPOOL | Restores files to a primary storage pool from copy storage pools.     |
| UPDATE STGPOOL  | Changes the attributes of a storage pool.                             |

## REPLY (Allow a request to continue processing)

Use this command and an identification number to inform the server that you have completed a requested operation. Not all server requests require a reply. This command is required only 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—request_number—┐
                           └LABEL—==—volume_label—┘
```

### Parameters

#### *request\_number* (Required)

Specifies the identification number of the request.

#### **LABEL**

Specifies the label to be written on a volume when you reply to a message from a LABEL LIBVOLUME command process. This parameter is optional.

### Example: Reply to a request

Respond to a reply request using 3 as the request number.

```
reply 3
```

### Related commands

Table 279. Commands related to REPLY

| Command        | Description  |
|----------------|--|
| CANCEL REQUEST | Cancels pending volume mount requests.                 |
| QUERY REQUEST  | Displays information about all pending mount requests. |

## RESET PASSEXP (Reset password expiration)

Use this command to reset the password expiration period to the common expiration period for administrator and client node passwords.

**Restriction:** You cannot reset the password expiration period to the common expiration period with the SET PASSEXP command.

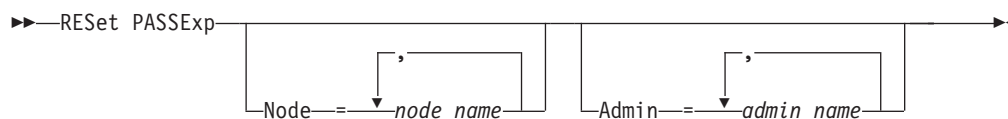
Use the QUERY STATUS command to display the common password expiration period.

**Restriction:** If you do not specify either the NODE or ADMIN parameters, the password expiration period for all client nodes and administrators will be reset.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### Node

Specifies the name of the node whose password expiration period you would like to reset. To specify a list of nodes, separate the names with commas and no intervening spaces. This parameter is optional.

#### Admin

Specifies the name of the administrator whose password expiration period you would like to reset. To specify a list of administrators, separate the names with commas and no intervening spaces. This parameter is optional.

### Example: Reset the password expiration for specific client nodes

Reset the password expiration period for client nodes bj and katie.

```
reset passexp node=bj,katie
```

### Example: Reset the password expiration for all users

Reset the password expiration period for all users to the common expiration period.

```
reset passexp
```

### Related commands

Table 280. Commands related to RESET PASSEXP

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## RESET PASSEXP

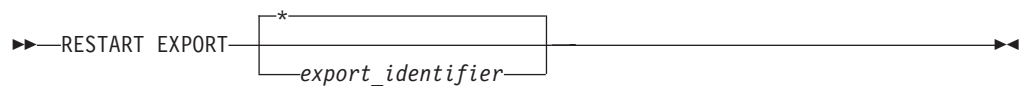
*Table 280. Commands related to RESET PASSEXP (continued)*

| Command      | Description   |
|--------------|---|
| SET PASSEXP  | Specifies the number of days after which a password is expired and must be changed. |
| UPDATE ADMIN | Changes the password or contact information associated with any administrator.      |
| UPDATE NODE  | Changes the attributes associated with a client node.                               |

An export operation is suspended when any of the following conditions is detected:

- Important:** Nodes or file spaces (on the exporting server) in the original export operation that are subsequently renamed are not included in the resumed operation. Any remaining data for nodes or file spaces on the target server that are deleted prior to resumption are discarded.

## Syntax



*export\_identifiser*

### Example: Restart a suspended export

Restart the suspended export operation identified by the export identifier EXPORTALLACCTNODES.

```
restart export exportallacctnodes
```

## Related commands

Table 281. Commands related to RESTART EXPORT

| Command       | Description   |
|---------------|---|
| CANCEL EXPORT | Deletes a suspended export operation                |
| EXPORT NODE   | Copies client node information to external media.   |
| EXPORT SERVER | Copies all or part of the server to external media. |

## RESTART EXPORT

*Table 281. Commands related to RESTART EXPORT (continued)*

| Command        | Description   |
|----------------|---|
| QUERY EXPORT   | Displays the export operations that are currently running or suspended. |
| SUSPEND EXPORT | Suspends a running export operation.                                    |

---

## RESTORE commands

Use the RESTORE commands to restore Tivoli Storage Manager storage pools or volumes.

The following is a list of RESTORE commands for Tivoli Storage Manager:

- “RESTORE NODE (Restore a NAS node)” on page 866
- “RESTORE STGPOOL (Restore storage pool data from a copy pool or an active-data pool)” on page 871
- “RESTORE VOLUME (Restore primary volume data from a copy pool or an active-data pool)” on page 875

## RESTORE NODE (Restore a NAS node)

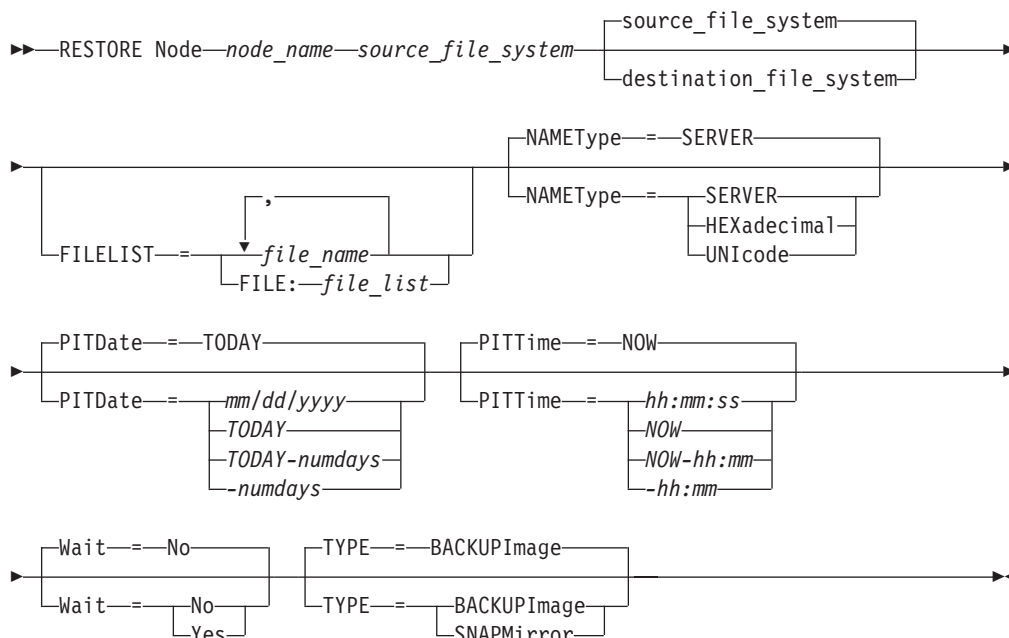
Use this command to initiate a restore operation for a network-attached storage (NAS) node.

You can use the `RESTORE NODE` command to restore backups that were created by using either the client's `BACKUP NAS` command or the server's `BACKUP NODE` command. NAS data may be restored from primary or copy native Tivoli Storage Manager pools; primary or copy NAS pools; or any combination needed to achieve the restore.

## Privilege class

To issue this command, you must have system privilege, policy privilege for the domain to which the node is assigned, or client owner authority over the node.

## Syntax



## Parameters

*node\_name* **(Required)**

Specifies the name of the node to restore. You cannot use wildcard characters or specify a list of names.

*source\_file\_system* (Required)

Specifies the name of the file system to restore. You cannot use wildcard characters for this name. You cannot specify more than one file system to restore. Virtual file space names are allowed.

*destination file system*

Specifies that the file server restores the data to an existing, mounted file system on the file server. This parameter is optional. The default is the original location of the file system on the file server. Virtual file space names are allowed.



**FILELIST**

Specifies the list of file or directory names to be restored. This parameter is optional. The default is to restore the entire file system. If this value is specified, the server attempts to restore the objects from the appropriate image. If the **PITDATE** and **PITTIME** parameters are specified, then the file is restored from the last backup image prior to the specified time. If no **PITDATE** and **PITTIME** parameters are specified, the file is restored from the latest backup image of the file system.

If the image is a differential backup, objects are first restored from the corresponding full backup and then from the differential backup. The restore is done by scanning the appropriate images for the specified objects and restoring any that are found. The TOCs for these images will not be accessed, so the IBM Tivoli Storage Manager server does not check whether the objects are actually contained within the images.

The folder path and file name must be entered using forward slash (/) symbols. However, no ending forward slash (/) is needed at the end of the file name.

All arguments that contain a space must have double quotation marks ("argument with spaces") surrounding the entire argument.

`FILELIST="/path/to/filename1 with blanks",/path/to/filename2_no_blanks`

Any file names that contain commas must have double quotation marks surrounding the entire argument, surrounded by single quotation marks ("argument with commas").

`FILELIST='"/path/to/filename1,with,commas",/path/to/filename2_no_commas`

*file\_name*

Specifies one or more file or directory names to be restored. The names you specify cannot contain wildcards. Multiple names must be separated with commas and no intervening blanks. File names are case-sensitive.

**FILE:***file\_list*

Specifies the name of a file that contains a list of the file or directory names to be restored. In the specified file, each file or directory name must be on a separate line. Blank lines and comment lines that begin with an asterisk are ignored. For example:

To restore files FILE01, FILE02, and FILE03, create a file named RESTORELIST that contains a line for each file:

```
FILE01
FILE02
FILE03
```

You can specify the files to be restored with the command as follows:

`FILELIST=FILE:RESTORELIST`

**NAMEType**

Specifies how you want the server to interpret the names specified as

**FILELIST**=*file\_name* or the names listed in the file specified with

**FILELIST**=*file\_list*. This parameter is useful when the names may contain

Unicode characters. It has no effect if the **FILELIST** parameter is not specified.

The default value is **SERVER**. Possible values are:

**SERVER**

The server uses the server's code page to interpret the names.

**HEXadecimal**

The server interprets the names that you enter as the hexadecimal

## RESTORE NODE

representation of a name in Unicode. To view the hexadecimal representation of a file or directory name, you can use the QUERY TOC command with FORMAT=DETAILED.

### UNICODE

The server interprets the names as being UTF-8 encoded. This option only applies when you have specified a list with FILELIST=FILE:file\_list.

**Restriction:** Network Data Management Protocol (NDMP) has limitations that prevent Tivoli Storage Manager from reporting whether or not individual files and directories are successfully restored.

### PITDate

Specifies the point-in-time date. When used with the PITTIME parameter, PITDATE establishes the point in time from which you want to select the data to restore. The latest data that was backed up on or before the date and time that you specify will be restored. This parameter is optional. The default is TODAY.

You can specify the date by using one of the following values:

| Value                     | Description                           | Example  |
|---------------------------|---------------------------------------|--|
| MM/DD/YYYY                | A specific date                       | 06/25/2001   |
| TODAY                     | The current date                      | TODAY  |
| TODAY-days<br>or<br>-days | The current date minus days specified | TODAY-7 or -7.<br><br>To restore data that was backed up a week ago, specify<br>PITDATE=TODAY-7 or PITDATE=-7. |

### PITTime

Specifies the point-in-time time. When used with the PITDATE parameter, PITTIME establishes the point in time from which you want to select the data to restore. The latest data that was backed up on or before the date and time that you specify will be restored. This parameter is optional. The default is the current time.

You can specify the time by using one of the following values:

| Value                     | Description  | Example  |
|---------------------------|--|--|
| HH:MM:SS                  | A specific time on the specified date                                | 12:33:28   |
| NOW                       | The current time on the specified date                               | NOW  |
| NOW-HH:MM<br>or<br>-HH:MM | The current time minus hours and minutes on the specified begin date | NOW-03:30 or -03:30.<br><br>If you issue this command at 9:00 with<br>PITTIME=NOW-03:30 or<br>PITTIME=-03:30, the server restores<br>backup records with a time of 5:30 or<br>later on the point-in-time date. |

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. The default is NO. Possible values are:

**No**

Specifies that the server processes this command in the background. Use the QUERY PROCESS command to monitor the background processing of this command.

**Yes**

Specifies that the server processes this command in the foreground. You wait for the command to complete before continuing with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Restriction:** You cannot specify WAIT=YES from the server console.

**TYPE**

Specifies the type of image to restore. The default value for this parameter is BACKUPIMAGE and it is used to restore data from standard NDMP base or differential backups. Other image types represent backup methods that might be specific to a particular file server. Possible values are:

**BACKUPImage**

Specifies that the file system should be restored from the appropriate standard NDMP backup images. This is the default method for performing an NDMP restore operation. Using the BACKUPIMAGE type, you can restore data from base and differential backups, and data at the file level.

**SNAPMirror**

Specifies that the file system should be retrieved from a NetApp SnapMirror image. SnapMirror images are block-level full-backup images of a NetApp file system. A SnapMirror image can only be restored to a file system that has been prepared as a SnapMirror target volume. Refer to the documentation that came with your NetApp file server for details.

After a SnapMirror image is retrieved and copied to a target file system, Tivoli Storage Manager breaks the SnapMirror relationship that was created by the file server during the operation. After the restore is complete, the target file system returns to the same state as that of the original file system at the point-in-time of the backup.

When setting the **TYPE** parameter to SNAPMIRROR, note the following restrictions:

**Restrictions:**

- You cannot specify the FILELIST parameter.
- Neither the *source\_file\_system\_name* nor the *destination\_file\_system\_name* can be a virtual file space name.
- This parameter is valid for NetApp and IBM N-Series file servers only.

**Example: Restore data from a file system**

Restore the data from the /vol/vol10 file system on NAS node NAS1.

```
restore node nas1 /vol/vol10
```

**Example: Restore a directory-level backup to the same location**

Restore the directory-level backup to the original location. The source is the virtual file space name /MIKESDIR and no destination is specified.

```
restore node nas1 /mikesdir
```

## RESTORE NODE

For this example and the next example, assume the following virtual file space definitions exist on the server for the node NAS1.

```
VFS Name  Filesystem Path
/mikesdir /vol/vol2 /mikes
/TargetDirVol2 /vol/vol2 /tmp
/TargetDirVol1 /vol/vol1 /tmp
```

### Example: Restore a directory-level backup to a different file system

Restore the directory-level backup to a different file system but preserve the path.

```
restore node nas1 /mikesdir /vol/vol0
```

### Related commands

*Table 282. Commands related to RESTORE NODE*

| Command                 | Description  |
|-------------------------|--|
| BACKUP NODE             | Backs up a network-attached storage (NAS) node.                            |
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping.                                       |
| QUERY NASBACKUP         | Displays information about NAS backup images.                              |
| QUERY TOC               | Displays details about the table of contents for a specified backup image. |

## RESTORE STGPOOL (Restore storage pool data from a copy pool or an active-data pool)

Use this command to restore files from one or more copy storage pools or active-data pools to a primary storage pool.

Tivoli Storage Manager restores all the primary storage pool files that:

- Have been identified as having errors
- Reside on a volume with an access mode of DESTROYED

You can also use this command to identify volumes that contain damaged, primary files. During restore processing, a message is issued for every volume in the restored storage pool that contains damaged, non-cached files. Use the QUERY CONTENT command to identify damaged, primary files on a specific volume.

You cannot restore a storage pool defined with a CENTERA device class.

In addition to restoring data to primary storage pools that have NATIVE or NONBLOCK data formats, this command also lets you restore data to primary storage pools that have NDMP data formats (NETAPPDUMP, CELERRADUMP, or NDMPDUMP). The primary storage pool must have the same data format as the copy storage pool from which data is to be restored. Tivoli Storage Manager supports backend data movement for NDMP images. For details, see the *Administrator's Guide*.

**Note:** To restore NAS client-node data to NAS storage pools, you must manually change the access mode of the volumes to DESTROYED using the UPDATE VOLUME command. However, if you are using disaster recovery manager, the plan file will contain the information the server needs to automatically mark the volumes as DESTROYED.

Restoration of files might be incomplete if backup file copies in copy storage pools or active-data pools were moved or deleted by other Tivoli Storage Manager processes during restore processing. To prevent this problem, do not issue the following commands for copy storage pool or active-data 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 percentage to 100 with the UPDATE STGPOOL command.

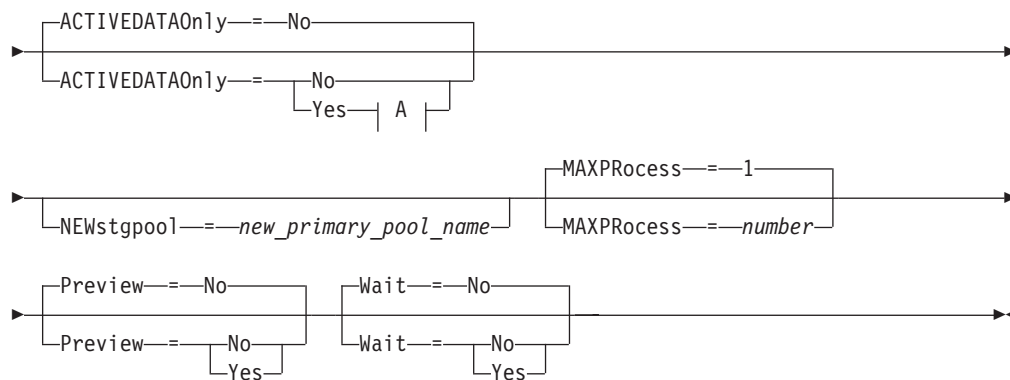
### Privilege class

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

```
►►—RESTORE STGpool—primary_pool_name—[COPYstgpool—=—copy_pool_name—]—►
```

## RESTORE STGPOOL



### A (Yes):

|—ACTIVEDATAPool—=*active-data\_pool\_name*—|

## Parameters

### *primary\_pool\_name* (Required)

Specifies the name of the primary storage pool that is being restored.

### COPYstgpool

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. Do not use this parameter with the ACTIVE DATA ONLY or ACTIVEDATAPool parameters.

### ACTIVEDATAOnly

Specifies that active versions of backup files are to be restored from active-data pools only. This parameter is optional. The default is NO. If this parameter is not specified, files are restored from copy-storage pools. Do not use this parameter with the COPYSTGPOOL parameter. Possible values are:

#### No

Specifies that the storage pool will not be restored from active-data pools.

#### Yes

Specifies that the storage pool will be restored from active-pool or pools that you specify using the ACTIVEDATAPool parameter. If you specify YES as a value for ACTIVE DATA ONLY, but do not specify a value for ACTIVEDATAPool, files are restored from any active-data pool in which active versions of backup files can be located.

**Attention:** Restoring a primary storage pool from an active-data pool might cause some or all inactive files to be deleted from the database if the server determines that an inactive file needs to be replaced but cannot find it in the active-data pool.

### ACTIVEDATAPool

Specifies the name of the active-data pool from which the active versions of backup files are to be restored. This parameter is optional. If this parameter is not specified, files are restored from any active-data pool in which active versions of backup files can be located.

### NEWstgpool

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**

Specifies the maximum number of parallel processes that are used for restoring files. Using multiple, parallel processes may improve throughput for the restore. This parameter is optional. You can specify a value from 1 to 999. The default is 1.

When determining this value, consider the number of mount points (logical drives) and physical drives that can be dedicated to this operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point, and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the restore.

Each process needs a mount point for copy storage pool volumes, and, if the device type is not FILE, each process also needs a drive. If you are restoring files in 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 a maximum of 3 processes to restore a primary sequential storage pool from a copy storage pool of the same device class. Each process requires two mount points and two drives. To run all three processes, the device class must have a mount limit of at least 6, and at least 6 mount points and 6 drives must be available.

To preview a restore, only one process is used and no mount points or drives are needed.

#### **Preview**

Specifies if you want to preview but not perform the restore. The preview lets you identify volumes required to restore the storage pool. The preview displays:

- A list of primary storage pool volumes that contain damaged files.
- The number of files and the number of bytes to 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 to be restored. These volumes must be mounted if you perform the restore.
- A list of any volumes containing files that cannot be restored.

**Note:** For only a list of offsite copy storage pool volumes to be mounted during a restore, change the access mode of the copy pool volumes to UNAVAILABLE. This prevents reclamation and move data processing of the volumes until they are moved onsite for the restore.

This parameter is optional. The default is NO. Possible values are:

#### **No**

Specifies that the restore is done.

#### **Yes**

Specifies that you want to preview the restore but not do the restore.

#### **Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is NO. Possible values are:

## RESTORE STGPOOL

### No

Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed.

Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged. To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files may have already been restored prior to the cancellation.

### Yes

Specifies that the server performs this operation in the foreground. The operation must complete before you can continue with other tasks. The server then displays the output messages to the administrative client when the operation completes.

**Note:** You cannot specify WAIT=YES from the server console.

### Example: Restore files from a copy storage pool to the primary storage pool

Restore files from any copy storage pool to the primary storage pool, PRIMARY\_POOL.

```
restore stgpool primary_pool
```

### Example: Restore files from a specific active-data pool to the primary storage pool

Restore files from active-data pool ADP1 to the primary storage pool PRIMARY\_POOL.

```
restore stgpool primary_pool activedataonly=yes activedatapool=adp1
```

## Related commands

*Table 283. Commands related to RESTORE STGPOOL*

| Command           | Description   |
|-------------------|---|
| BACKUP STGPOOL    | Backs up a primary storage pool to a copy storage pool.                                       |
| CANCEL PROCESS    | Cancels a background server process.  |
| COPY ACTIVATEDATA | Copies active backup data.  |
| QUERY CONTENT     | Displays information about files in a storage pool volume.                                    |
| QUERY PROCESS     | Displays information about background processes.  |
| RESTORE VOLUME    | Restores files stored on specified volumes in a primary storage pool from copy storage pools. |
| UPDATE STGPOOL    | Changes the attributes of a storage pool.   |
| UPDATE VOLUME     | Updates the attributes of storage pool volumes.   |



## RESTORE VOLUME (Restore primary volume data from a copy pool or an active-data pool)

Use this command to restore all files on damaged volumes in a primary storage pool that was backed up to a copy storage pool or copied to an active-data pool. Tivoli Storage Manager does not restore cached copies of files and removes those cached files from the database during restore processing.

In addition to restoring data to volumes in storage pools that have NATIVE or NONBLOCK data formats, this command also lets you restore data to volumes in storage pools that have NDMP data formats (NETAPPDUMP, CELERRADUMP, or NDMPDUMP). The volumes to be restored must have the same data format as the volumes in the copy storage pool. Tivoli Storage Manager supports backend data movement for NDMP images. For details, see the *Administrator's Guide*.

This command changes the access mode of the specified volumes to DESTROYED. When all files on a volume are restored to other locations, the destroyed volume is empty and is 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 processes during a restore. See note 3.
- An active-data pool was specified for the restore, and inactive files were not available to be copied.

### Important:

1. You cannot restore volumes in storage pools defined with a CENTERA device class.
2. Before you restore a random-access volume, issue the VARY command to vary the volume offline.
3. To prevent copy storage pools files from being moved or deleted by other processes, do not issue the following commands for copy storage pool volumes during a restore:
  - MOVE DATA
  - DELETE VOLUME (DISCARDATA=YES)
  - AUDIT VOLUME (FIX=YES)

To prevent reclamation processing of copy storage pools, issue the UPDATE STGPOOL command with the RECLAIM parameter set to 100.

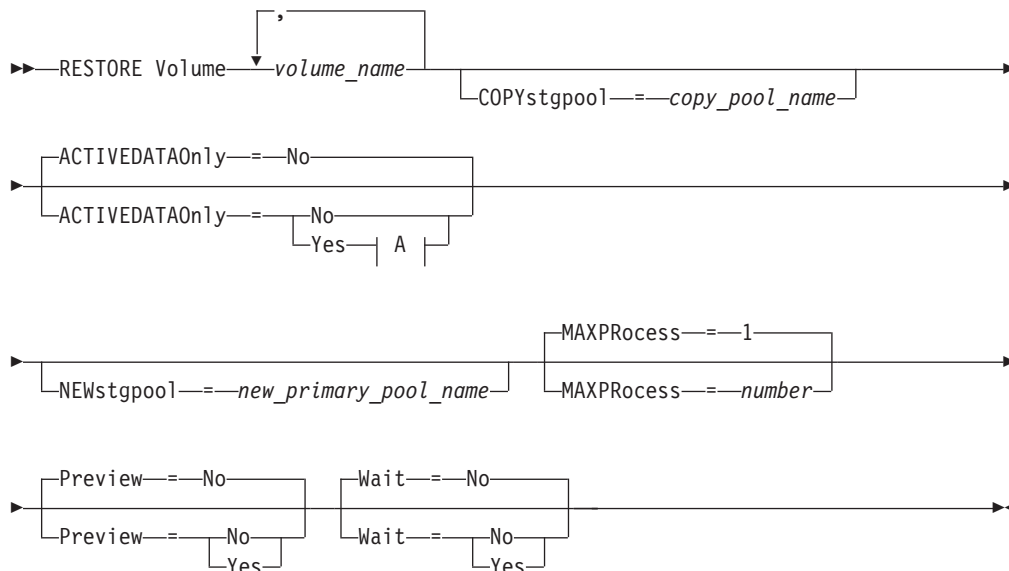
### Privilege class

To issue this command you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the primary storage pool. If you have restricted privilege and want to restore files to a new primary storage pool, you

## RESTORE VOLUME

must also have authority for the new storage pool.

### Syntax



### A (Yes):

[ACTIVEDATAPool=active-data\_pool\_name]

### Parameters

#### *volume\_name* (Required)

Specifies the name of the primary storage pool volume to be restored. To specify a list of volumes that belong to the same primary storage pool, separate the names with commas and no intervening spaces.

#### **COPYstgpool**

Specifies the name of the copy storage pool from which the files are to be restored. This parameter is optional. If you do not specify this parameter, files are restored from any copy pool in which copies can be located. Do not use this parameter with the **ACTIVEDATAONLY** or **ACTIVEDATAPOOL** parameters.

#### **ACTIVEDATAOnly**

Specifies that active versions of backup files are to be restored from active-data pools only. This parameter is optional. The default is **NO**. If this parameter is not specified, files are restored from copy-storage pools. Do not use this parameter with the **COPYSTGPOOL** parameter. Possible values are:

##### **No**

Specifies that the storage pool will not be restored from active-data pools.

##### **Yes**

Specifies that the storage pool will be restored from active-pool or pools that you specify using the **ACTIVEDATAPOOL** parameter. If you specify **YES** as a value for **ACTIVEDATAONLY**, but do not specify a value for

ACTIVEDATAPOOL, files are restored from any active-data pool in which active versions of backup files can be located.

**Attention:** Restoring a volume from an active-data pool might cause some or all inactive files to be deleted from the database if the server determines that an inactive file needs to be replaced but cannot find it in the active-data pool.

#### **ACTIVEDATAPool**

Specifies the name of the active-data pool from which the active versions of backup files are to be restored. This parameter is optional. If this parameter is not specified, files are restored from any active-data pool in which active versions of backup files can be located.

#### **NEWstgpool**

Specifies the name of the new storage pool to which to restore the files. This parameter is optional. If you do not specify this parameter, files are restored to the original primary storage pool.

#### **MAXProcess**

Specifies the maximum number of parallel processes to use for restoring files. Using parallel processes may improve throughput. This parameter is optional. You can specify a value from 1 to 999. The default is 1.

When determining this value, consider the number of mount points (logical drives) and physical drives that can be dedicated to this operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point, and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the restore.

Each process needs a mount point for copy storage pool volumes. If the device type is not FILE, each process also needs a drive. If you are restoring a sequential storage pool, each process needs an additional mount point for primary storage pool volumes and, if the device type is not FILE, an additional drive. For example, suppose you specify a maximum of three processes to back up a primary sequential storage pool to a copy storage pool of the same device class. Each process requires two mount points and two drives. To run all three processes, the device class must have a mount limit of at least 6, and at least 6 mount points and 6 drives must be available.

To preview a backup, only one process is used and no mount points or drives are needed.

#### **Preview**

Specifies if you want to preview but not perform the restore. You can use this option to identify the offsite volumes required to restore a storage pool. This parameter is optional. The default 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 but restore the data.

**Tip:** If you preview a restore to see a list of offsite copy pool volumes to be mounted, you should you change the access mode of the identified

## RESTORE VOLUME

volumes to UNAVAILABLE. This prevents reclamation and MOVE DATA processing for these volumes until they are transported to the onsite location for use in restore processing.

The preview displays the following:

- The number of files and bytes to be restored, if the access mode of the copy storage pool volumes is READWRITE or READONLY when the restoration is performed.
- A list of copy storage pool volumes containing files to be restored. These volumes must be mounted if you perform the restore.
- A list of volumes containing files that cannot be restored.

### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. This default is NO. Possible values are:

#### No

Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed. Messages created from the background process are displayed either in the activity log or the server console, depending on where messages are logged.

To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files may have already been backed up prior to the cancellation.

#### Yes

Specifies that the server processes this command in the foreground. The operation must complete before you can continue with other tasks. The server then displays the output messages to the administrative client when the command completes.

**Remember:** You cannot specify WAIT=YES from the server console.

### Example: Restore primary volume data files

Restore files stored on volume PVOL2 in primary storage pool PRIMARY\_POOL.

```
restore volume pv02
```

### Example: Restore primary volume data files from an active-data pool

Restore files stored on volume VOL001 in primary pool PRIMARY\_POOL from active-data pool ADP1.

```
restore volume vol001 activedataonly=yes activedatapool=adp1
```

### Related commands

Table 284. Commands related to RESTORE VOLUME

| Command           | Description   |
|-------------------|---|
| BACKUP STGPOOL    | Backs up a primary storage pool to a copy storage pool. |
| COPY ACTIVATEDATA | Copies active backup data.                              |
| CANCEL PROCESS    | Cancels a background server process.                    |

*Table 284. Commands related to RESTORE VOLUME (continued)*

| Command         | Description   |
|-----------------|---|
| QUERY PROCESS   | Displays information about background processes.                  |
| RESTORE STGPOOL | Restores files to a primary storage pool from copy storage pools. |

### REVOKE commands

Use the REVOKE command to revoke one or more privilege classes from an administrator for Tivoli Storage Manager.

The following is a list of REVOKE commands for Tivoli Storage Manager:

- “REVOKE AUTHORITY (Remove administrator authority)” on page 881
- “REVOKE PROXYNODE (Revoke proxy authority for a client node)” on page 885

# REVOKE AUTHORITY (Remove administrator authority)

Use this 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 and the number of storage pools to which a restricted storage administrator has authority.

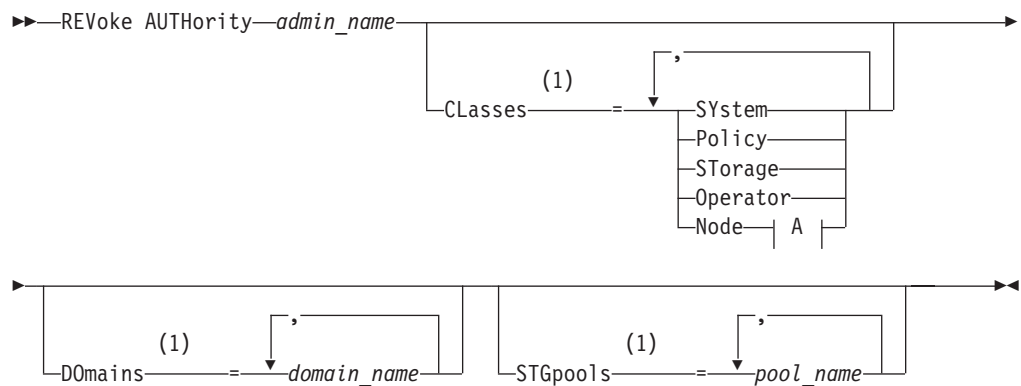
If you use the REVOKE AUTHORITY command without the CLASSES, DOMAINS, and STGPOLLS parameters, you will revoke all privileges for the specified administrator.

At least one administrator must have system privilege; therefore, if the administrator is the only one with system privilege, you cannot revoke the authority.

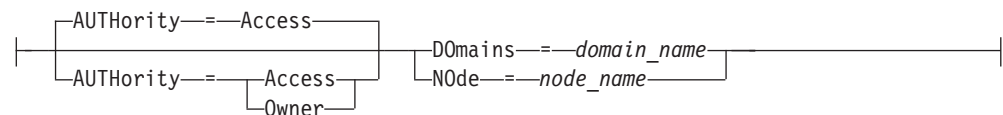
## Privilege class

To issue this command, you must have system privilege.

## Syntax



**A:**



## Notes:

- 1 If all these parameters are omitted, all administrator privileges will be revoked for this administrator.

## Parameters

### admin\_name (Required)

Specifies the name of the administrator whose administrative privilege is to be revoked or reduced.

### Classes

Specifies one or more administrative privilege classes to be revoked. You can specify more than one class by separating 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.

### **N**ode

Indicates that node privilege is to be revoked for this user.

### **A**UTHority

Indicates the authority level to revoke for a user with node privilege. This parameter is optional.

If an administrator already has system or policy privilege to the policy domain to which the node belongs, this command will not change the administrator's privilege. Possible authority levels are:

#### **Access**

Indicates that client access authority is revoked. This is the default when CLASSES=NODE is specified.

**Note:** A client node can set the REVOKEREMOTEACCESS option to prevent access by a user with node privilege and client access authority. If a user with node privilege has client owner authority, or has system or policy privileges to the policy domain to which the node belongs, that administrator can still access the web backup-archive client.

#### **Owner**

Indicates that client owner authority is revoked.

### **D**Omains

Indicates that you want to revoke an administrator's client access or client owner authority to all clients in the specified policy domain. This parameter cannot be used together with the NODE parameter.

### **N**ode

Indicates that you want to revoke an administrator's client access or client owner authority to the node. This parameter cannot be used together with the DOMAIN parameter.

### **D**Omains

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 wildcard characters to specify a name. Authority for all matching domains will be revoked. If DOMAINS is specified, the parameter CLASSES=POLICY is optional.



## STGpools

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 wildcard characters 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. 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.

To revoke unrestricted storage privilege from an administrator, specify the CLASSES=STORAGE parameter. You cannot use the STGPOOLS parameter to revoke authority for selected storage pools from an unrestricted storage administrator.

2. 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 revoke unrestricted policy privilege from an administrator, specify the CLASSES=POLICY parameter. You cannot use the DOMAINS parameter to revoke authority for selected domains from an unrestricted administrator.

## Example: Revoke certain administrative privileges

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.

```
revoke authority claudia classes=policy
domains=employee_records
```

## Example: Revoke all administrative privileges

Administrator LARRY currently has operator 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 STGPOOLS. LARRY remains an administrator but he can only use those commands that can be issued by any administrator.

```
revoke authority larry
```

## Example: Revoke node privilege

Help desk personnel user CONNIE currently has node privilege with client owner authority for client node WARD3. Revoke her node privilege with client owner authority.

```
revoke authority connie classes=node
authority=owner node=ward3
```

## REVOKE AUTHORITY

### Related commands

*Table 285. Commands related to REVOKE AUTHORITY*

| Command         | Description   |
|-----------------|---|
| GRANT AUTHORITY | Assigns privilege classes to an administrator.                                    |
| QUERY ADMIN     | Displays information about one or more IBM Tivoli Storage Manager administrators. |

## REVOKE PROXYNODE (Revoke proxy authority for a client node)

Use this command to revoke authority for an agent client node to perform backup and restore operations for a target node on the Tivoli Storage Manager server.

### Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege

### Syntax

```
►►—REVoke PROXynode TArget—==—target_node_name—AGent—==—agent_node_name—►◄
```

### Parameters

#### TArget (Required)

Specifies the target node to which an agent node has been granted proxy authority. Wildcard characters and comma-separated lists of node names are allowed.

#### AGent (Required)

Specifies which node has authority to act as proxy to the target node. Wildcard characters and comma-separated lists of node names are allowed.

### Example: Revoke a node's proxy authority

To revoke authority from target node NASCLUSTER to act as proxy for all agent nodes which start with the letter M, issue the following command.

```
revoke proxynode target=nascluster agent=m*
```

### Related commands

*Table 286. Commands related to REVOKE PROXYNODE*

| Command         | Description   |
|-----------------|---|
| GRANT PROXYNODE | Grant proxy authority to an agent node.             |
| QUERY PROXYNODE | Display nodes with authority to act as proxy nodes. |

## ROLLBACK (Rollback uncommitted changes in a macro)

Use this command within a macro to undo any processing changes made by commands run by the server but not yet committed to the database. A committed change is permanent and cannot be rolled back. The ROLLBACK command is useful for testing macros.

Ensure that your administrative client session is not running with the ITEMCOMMIT option when using this command.

**Important:** SETOPT commands inside a macro cannot be rolled back.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—ROLLBACK—◄◄

### Parameters

None.

### Example: Rollback changes in a macro

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 and 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 */
```

### Related commands

Table 287. Commands related to ROLLBACK

| Command | Description                              |
|---------|--|
| COMMIT  | Makes changes to the database permanent. |
| MACRO   | Runs a specified macro file.             |

## RUN (Run a Tivoli Storage Manager script)

Use this command to run a Tivoli Storage Manager script. To issue this command on another server, the script being run must be defined on that server.

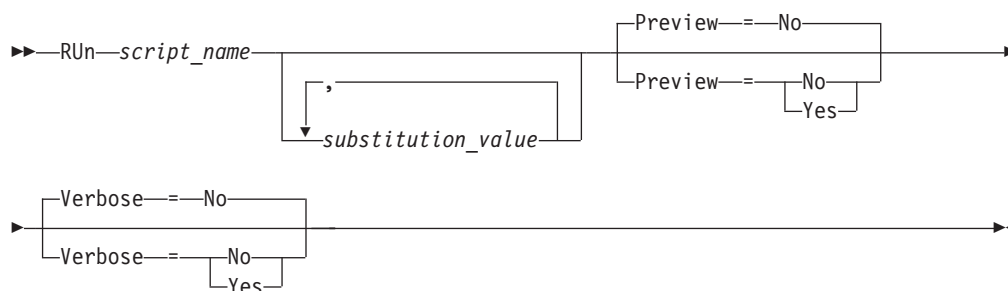
You can include RUN commands in scripts as long as they do not create loops. For example, you should avoid including RUN commands where SCRIPT\_A runs SCRIPT\_B and SCRIPT\_B runs SCRIPT\_A.

**Important:** Tivoli Storage Manager does not have a command that can cancel a script after it starts. To stop a script, you must halt the server.

### Privilege class

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

### Syntax



### Parameters

#### *script\_name* (Required)

Specifies the name of the script you want processed. The name you specify cannot be a substitution variable, such as \$1.

#### *substitution\_value*

Specifies one or more values to substitute for variables when the script is run. In a script, a substitution variable consists of a '\$' character, followed by a number. When you run the script, Tivoli Storage Manager replaces the substitution variables defined in a script with the values you supply with this command. You must specify values for each substitution variable defined in the script or the script will fail. This parameter is optional.

#### Preview

Specifies whether to preview the command lines of a script without actually processing the script. The default is NO.

Possible values are:

##### Yes

Specifies that the command lines included in a script are displayed, but the script is not processed.

##### No

Specifies that the command lines included in a script are displayed and the script is processed.

## RUN

### Verbose

Specifies whether command lines, variable substitution, and conditional logic testing used in a script are displayed as the script is being processed. This parameter is ignored if PREVIEW=YES is specified. The default is NO.

Possible values are:

#### Yes

Specifies that the command lines, variable substitution, and conditional logic testing are displayed as the script is being processed.

#### No

Specifies that the command lines, variable substitution, and conditional logic testing do not display as the script is being processed.

### Example: View the commands generated by a script with a table name substitution variable

To run the following example script, called QSAMPLE, you issue a RUN command that specifies the table name ACTLOG as the value for the substitution variable, \$1. Use the output to preview the commands generated by the script before running the commands.

```
001  /* This is a sample SQL Query in wide format */
005  SET SQLDISPLAYMODE WIDE
010  SELECT colname FROM -
015  COLUMNS WHERE TABNAME='$1'

run qsample actlog preview=yes
```

```
ANR1461I RUN: Executing command script QSAMPLE.
ANR1466I RUN: Command script QSAMPLE, Line 5 :
           set sqldisplaymode wide.
ANR1466I RUN: Command script QSAMPLE, Line 15 :
           select colname from columns where tablename='ACTLOG'.
ANR1470I RUN: Command script QSAMPLE completed successfully
           (PREVIEW mode)
```

### Example: Run a script to display and run the commands generated by the script

Run the same script as show in the prior example to display both the generated commands and the results of the commands.

```
run qsample actlog verbose=yes
```

```

ANR1461I RUN: Executing command script QSAMPLE.
ANR1466I RUN: Command script QSAMPLE, Line 5 :
               set sqldisplaymode wide.
ANR1466I RUN: Command script QSAMPLE, Line 5 : RC=RC_OK
ANR1466I RUN: Command script QSAMPLE, Line 15 :
               select colname from columns where tabname='ACTLOG'.

```

```

COLNAME
-----
DATE_TIME
MSGNO
SEVERITY
MESSAGE
ORIGINATOR
NODENAME
OWNERNAME
SCHEDNAME
DOMAINNAME
SESSID

```

```

ANR1462I RUN: Command script QSAMPLE, Line 15 : RC=RC_OK
ANR1462I RUN: Command script QSAMPLE completed successfully.

```

### Example: Run a script to display just the results of the commands in the script

Run the previous example script, without displaying just the results of the generated commands in the script.

```
run qsampl actlog verbose=no
```

```

COLNAME
-----
DATE_TIME
MSGNO
SEVERITY
MESSAGE
ORIGINATOR
NODENAME
OWNERNAME
SCHEDNAME
DOMAINNAME
SESSID

```

```
ANR1462I RUN: Command script QSAMPLE completed successfully.
```

### Related commands

*Table 288. Commands related to RUN*

| Command       | Description  |
|---------------|--|
| COPY SCRIPT   | Creates a copy of a script.                                |
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| DELETE SCRIPT | Deletes the script or individual lines from the script.    |
| QUERY SCRIPT  | Displays information about scripts.                        |
| RENAME SCRIPT | Renames a script to a new name.                            |
| UPDATE SCRIPT | Changes or adds lines to a script.                         |

---

# SELECT (Perform an SQL query of the IBM Tivoli Storage Manager database)

Use this command to create and format a customized query of the IBM Tivoli Storage Manager database.

Tivoli Storage Manager Version 6.1 provides a SQL interface to a DB2 program. Restrictions and guidelines for handling SQL queries are handled directly by DB2.

To help you find what information is available, Tivoli Storage Manager provides three system catalog tables:

### **SYSCAT.TABLES**

Contains information about all tables that can be queried with the SELECT command.

### **SYSCAT.COLUMNS**

Describes the columns in each table.

### **SYSCAT.ENUMTYPES**

Defines the valid values for each type and the ordering of those values for columns that have an enumerated data type (SQL93).

You can issue the SELECT command to query these tables to determine the location of the information that you want.

## **Usage notes**

You cannot issue the SELECT command from a server console.

Because the select command does not lock and unlock records, contention for a record can cause the server to erroneously issue message ANR2034E: SELECT: No match found using this criteria. Check your selection criteria, and if you believe it is correct, try the command again.

To stop the processing of a SELECT command after it starts, cancel the administrative session from which the command was issued. Cancel the session from either the server console or another administrative session.

Temporary tablespaces are used to process SQL queries within DB2. Inadequate temporary space may cause SQL queries to fail.

## **Privilege class**

Any administrator can issue this command.

## **Syntax**

For SELECT statement syntax and guidelines, refer to the DB2 Information Center: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>. Search on *Select-statement* for usage guidelines.

## **List of examples**

The SELECT command lets you customize a wide variety of queries. To give you an idea of what you can do with the command, this section includes many



examples. There are, however, many more possibilities. Query output is shown only for the more complex commands to illustrate formatting.

The following list summarizes the example SELECT commands:

- List available tables
- List client nodes and administrative clients that are currently locked from server access
- List client nodes and administrative clients that have not specified the correct password lately
- List nodes in the standard policy domain that are not associated with the daily backup schedule DAILYBACKUP
- List the administrators that have policy authority
- List type E (ERROR) or W (WARNING) messages that have been issued in the time period for which activity log records have been maintained
- List the administrative schedules that have been defined or altered by administrator JAKE
- List the relative administrative schedule priorities
- List the management classes that have an archive copy group with a retention period greater than 365 days
- List the client nodes that are in each policy domain
- Count how many files have been archived from each node
- List the clients that are using space management
- Determine how many volumes would be reclaimed if the reclamation threshold were to be changed to 50 percent for storage pool TAPE
- Determine how many backup files would be affected for each node if the DAILY management class in the STANDARD policy domain is changed or deleted
- For all active client sessions, determine how long have they been connected and their effective throughput in bytes per second
- Determine how long the current background processes have been running and determine their effective throughput in time and files per second
- Count the number of client nodes are there for each platform type
- Count the number of file spaces each client node has and list the client nodes ascending order
- Obtain statistical information for calculating the number of offsite volumes that will have their space reclaimed during reclamation of a storage pool.
- Obtain statistical information for all export operations
- Obtain statistical information for the 'EXPORTNODE0001' export operation
- Obtain process identifier for export operations that are running

### Example: List available tables

List all the tables available for querying the IBM Tivoli Storage Manager database.

```
select * from syscat.tables
```

## SELECT

```
TABSCHEMA: SERVER1
TABNAME: ACTLOG
CREATE_TIME: 1999-05-01 07:39:06
COLCOUNT: 10
INDEX_COLCOUNT: 1
UNIQUE_INDEX: FALSE
REMARKS: Server activity log

TABSCHEMA: SERVER1
TABNAME: ADMIN_SCHEDULES
CREATE_TIME: 1995-05-01 07:39:06
COLCOUNT: 14
INDEX_COLCOUNT: 1
UNIQUE_INDEX: TRUE
REMARKS: Administrative command schedules

TABSCHEMA: SERVER1
TABNAME: ADMINS
CREATE_TIME: 1995-05-01 07:39:06
COLCOUNT: 15
INDEX_COLCOUNT: 1
UNIQUE_INDEX: TRUE
REMARKS: Server administrators

TABSCHEMA: SERVER1
TABNAME: ARCHIVES
CREATE_TIME: 1995-05-01 07:39:06
COLCOUNT: 10
INDEX_COLCOUNT: 5
UNIQUE_INDEX: FALSE
REMARKS: Client archive files
```

### Example: List client nodes and administrative clients that are currently locked from server access

```
select node_name from nodes where locked='YES'
```

```
select admin_name from admins where locked='YES'
```

### Example: List client nodes and administrative clients that have not specified the correct password lately

```
select node_name from nodes where invalid_pw_count <>0
```

```
select admin_name from admins where invalid_pw_count <>0
```

### Example: List nodes in the standard policy domain that are not associated with the daily backup schedule DAILYBACKUP

```
select node_name from nodes where domain_name='STANDARD' and
node_name not in (select node_name from associations
where domain_name='STANDARD' and
schedule_name='DAILYBACKUP')
```

### Example: List the administrators that have policy authority

```
select admin_name from admins where
upper(system_priv) <>'NO'
or upper(policy_priv) <>'NO'
```

### Example: List type E (ERROR) or W (WARNING) messages that have been issued in the time period for which activity log records have been maintained

```
select date_time,msgno,message from actlog
where severity='E' or severity='W'
```

**Example: List the administrative schedules that have been defined or altered by administrator JAKE**

```
select schedule_name from admin_schedules
where chg_admin='JAKE'
```

**Example: List the relative administrative schedule priorities**

```
select schedule_name,priority from admin_schedules order
by priority
```

**Example: List the management classes that have an archive copy group with a retention period greater than 365 days**

```
select domain_name,set_name,class_name from ar_copygroups
where retver='NOLIMIT' or cast(retver as integer) >365
```

**Example: List the management classes that specify more than 5 backup versions**

```
select domain_name,set_name,class_name from bu_copygroups
where verexists ='NOLIMIT' or
cast(verexists as integer)>5
```

**Example: List the client nodes that are using the client option set named SECURE**

```
select node_name from nodes where option_set='SECURE'
```

**Example: List the client nodes that are in each policy domain**

```
select domain_name,num_nodes from domains
```

**Example: Count how many files have been archived from each node**

**Attention:** This command could take a long time to complete.

```
select node_name,count(*) from archives
group by node_name
```

**Example: List the clients that are using space management**

```
select node_name from auditocc where spacemg_mb <>0
```

**Example: Determine how many volumes would be reclaimed if the reclamation threshold were to be changed to 50 percent for storage pool TAPE**

```
select count(*) from volumes where stgpool_name='TAPE'
and upper(status)='FULL' and pct_utilized < 50
```

**Example: Determine how many backup files would be affected for each node if the DAILY management class in the STANDARD policy domain is changed or deleted**

**Note:** This command will take significant time and resources to complete.

```
select node_name, count(*) as "Files" from backups
where class_name='DAILY' and node_name in
(select node_name from nodes where domain_name='STANDARD')
group by node_name
```

## SELECT

**Example: For all active client sessions, determine how long have they been connected and their effective throughput in bytes per second**

```
select session_id as "Session",
client_name as "Client",
state as "State",
current_timestamp-start_time as "Elapsed Time",
((cast(bytes_sent as decimal(18,0)) /
cast(second(current_timestamp-start_time) as decimal(18,0)))
as "Bytes sent/second",
((cast(bytes_received as decimal(18,0)) /
cast(second(current_timestamp-start_time) as decimal(18,0)))
as "Bytes received/second"
from sessions
```

```
Session: 24
Client: ALBERT
State: Run
Elapsed Time: 0 01:14:05.000000
Bytes sent/second: 564321.9302768451
Bytes received/second: 0.0026748857944
```

```
Session: 26
Client: MILTON
State: Run
Elapsed Time: 0 00:06:13.000000
Bytes sent/second: 1638.5284210992221
Bytes received/second: 675821.6888561849
```

**Example: Determine how long the current background processes have been running and determine their effective throughput in time and files per second**

**Note:** Expiration does not report the number of bytes processed.

```
select process_num as "Number",
process,
current_timestamp-start_time as "Elapsed Time",
((cast(files_processed as decimal(18,0)) /
cast(second(current_timestamp-start_time) as decimal(18,0)))
as "Files/second",
((cast(bytes_processed as decimal(18,0)) /
cast(second(current_timestamp-start_time) as decimal(18,0)))
as "Bytes/second"
from processes
```

```
Number: 1
PROCESS: Expiration
Elapsed Time: 0 00:24:36.000000
Files/second: 6.3216755870092
Bytes/second: 0.0000000000000
```

**Example: Count the number of client nodes for each platform type**

```
select platform_name,count(*) as "Number of Nodes"
from nodes group by platform_name
```

| PLATFORM_NAME | Number of Nodes |
|---------------|-----------------|
| AIX           | 6               |
| SunOS         | 27              |
| Win32         | 14              |
| Linux         | 20              |

**Example: Count the number of file spaces each client node has and list the client nodes ascending order**

```
select node_name, count(*) as "number of filespaces"
from filespaces group by node_name order by 2
```

| NODE_NAME | number of filespaces |
|-----------|----------------------|
| ALBERT    | 2                    |
| MILTON    | 2                    |
| BARNEY    | 3                    |
| SEBASTIAN | 3                    |
| MAILHOST  | 4                    |
| FALCON    | 4                    |
| WILBER    | 4                    |
| NEWTON    | 4                    |
| JEREMY    | 4                    |
| WATSON    | 5                    |
| RUSSELL   | 5                    |

**Example: Obtain statistical information for calculating the number of offsite volumes that will have their space reclaimed during reclamation of a storage pool.**

```
select * from summary where activity='OFFSITE RECLAMATION'
```

```

START_TIME: 2004-06-16 13:47:31.000000
END_TIME: 2004-06-16 13:47:34.000000
ACTIVITY: OFFSITE RECLAMATION
NUMBER: 4
ENTITY: COPYPOOL
COMMMETH:
ADDRESS:
SCHEDULE_NAME:
EXAMINED: 170
AFFECTED: 170
FAILED: 0
BYTES: 17821251
IDLE: 0
MEDIAM: 0
PROCESSES: 2
SUCCESSFUL: YES
VOLUME_NAME:
DRIVE_NAME:
LIBRARY_NAME:
LAST_USE:
COMM_WAIT:
NUM_OFFSITE_VOLS: 2
```

## SELECT

### Example: Obtain statistical information for all export operations

```
select * from export_operations
```

```
EXPORT IDENTIFIER: MyExportNode
STATUS: RUNNING
COMMAND: Export NODE m* filespace=c$
        nametype=unicode
        filedata=all toserver=-athens
START TIME: 2007-01-24 09:45:06.000000
PROCESS ID: 35
DATE AND TIME OF LAST RESTART: 2007-01-27 13:01:06.000000
PROCESSING DURATION (MINUTES): 120
TOTAL EXECUTION MINUTES: 457
TOTAL BYTES TRANSFERRED (MB): 7,000
```

### Example: Obtain statistical information for the 'EXPORTNODE0001' export operation

```
select command from export_operations
where export_identifier='EXPORTNODE0001'
```

```
COMMAND_NAME: Export NODE m* filespace=c$
nametype=unicode filedata=all toserver=athens
```

### Example: Obtain process identifier for export operations that are running

```
select process_id from export_operations where status='RUNNING'
```

```
PROCESS_ID
-----
35
```

## SERIAL (Run multiple commands in a script in serial)

Use this command in a script to ensure that any preceding commands are complete before proceeding and to ensure any following commands are run serially.

When a script starts, all commands run serially until a PARALLEL command is encountered.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—SERIAL—◄◄

### Parameters

None.

### Example: Write a script using the SERIAL command

Write a script named BACKUP to back up two storage pools simultaneously. When both backups are complete, perform a snapshot database backup. Issue the run backup command to run the script.

```
/* set up for running parallel commands */
PARALLEL
/* backup two storage pools simultaneously */
BACKUP STGPPOOL PRIMPOOL1 COPYPOOL1 WAIT=YES
BACKUP STGPPOOL PRIMPOOL2 COPYPOOL2 WAIT=YES
/* wait for all previous commands to finish and set up
running serial commands*/
SERIAL
/* start snapshot db backup */
BACKUP DB DEVCLASS=TAPE TYPE=DBSNAPSHOT WAIT=YES
```

### Related commands

Table 289. Commands related to SERIAL

| Command       | Description  |
|---------------|--|
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| RUN           | Runs a script.   |
| PARALLEL      | Run commands in a script in parallel.                      |

---

## SET commands

Use the SET command to select or specify a value for a Tivoli Storage Manager object.

The following is a list of SET commands for Tivoli Storage Manager:

- “SET ACCOUNTING (Set accounting records on or off)” on page 900
- “SET ACTLOGRETENTION (Set the retention period or the size of the activity log)” on page 901
- “SET ARCHIVERETENTIONPROTECTION (Activate data retention protection)” on page 903
- “SET AUTHENTICATION (Set password authentication)” on page 905
- “SET CLIENTACTDURATION (Set the duration period for the client action)” on page 906
- “SET CONFIGMANAGER (Specify a configuration manager)” on page 907
- “SET CONFIGREFRESH (Set managed server configuration refresh)” on page 908
- “SET CONTEXTMESSAGING (Set message context reporting on or off)” on page 909
- “SET CROSSDEFINE (Specifies whether to cross-define servers)” on page 910
- “SET DBRECOVERY (Set the device class for automatic backups)” on page 911
- “SET DBREPORTMODE (Set the level of database reporting)” on page 912
- “SET DRMACTIVEDATASTGPOOL (Specify the active-data pools to be managed by DRM)” on page 913
- “SET DRMCHECKLABEL (Specify label checking)” on page 915
- “SET DRMCMDFILENAME (Specify the name of a file to contain commands)” on page 916
- “SET DRMCOPYSTGPOOL (Specify the copy storage pools to be managed by DRM)” on page 917
- “SET DRMCOURIERNAME (Specify the courier name)” on page 918
- “SET DRMDBBACKUPEXPIREDAYS (Specify DB backup series expiration)” on page 919
- “SET DRMFILEPROCESS (Specify file processing)” on page 921
- “SET DRMINSTRPREFIX (Specify the prefix for recovery instructions file names)” on page 922
- “SET DRMNOTMOUNTABLENAME (Specify the not mountable location name)” on page 924
- “SET DRMPLANPREFIX (Specify a prefix for recovery plan file names)” on page 925
- “SET DRMPLANVPOSTFIX (Specify replacement volume names)” on page 927
- “SET DRMPRIMSTGPOOL (Specify the primary storage pools to be managed by DRM)” on page 928
- “SET DRMRPFEXPIREDAYS (Set criteria for recovery plan file expiration)” on page 929
- “SET DRMVAULTNAME (Specify the vault name)” on page 931
- “SET EVENTRETENTION (Set the retention period for event records)” on page 932
- “SET INVALIDPWLIMIT (Set the number of invalid logon attempts)” on page 933



- “SET LICENSEAUDITPERIOD (Set license audit period)” on page 934
- “SET MAXCMDRETRIES (Set the maximum number of command retries)” on page 935
- “SET MAXSCHEDSESSIONS (Set maximum scheduled sessions)” on page 936
- “SET MINPWLENGTH (Set minimum password length)” on page 937
- “SET PASSEXP (Set password expiration date)” on page 938
- “SET QUERYSCHEDPERIOD (Set query period for polling client nodes)” on page 940
- “SET RANDOMIZE (Set randomization of scheduled start times)” on page 941
- “SET REGISTRATION (Set open or closed registration)” on page 943
- “SET RETRYPERIOD (Set time between retry attempts)” on page 945
- “SET SCHEDMODES (Select a central scheduling mode)” on page 946
- “SET SERVERHLADDRESS (Set the high-level address of a server)” on page 948
- “SET SERVERLLADDRESS (Set the low-level address of a server)” on page 949
- “SET SERVERNAME (Specify the server name)” on page 950
- “SET SERVERPASSWORD (Set password for server)” on page 951
- “SET SUBFILE (Set subfile backup for client nodes)” on page 952
- “SET SUMMARYRETENTION (Set number of days to keep data in activity summary table)” on page 953
- “SET TAPEALERTMSG (Set tape alert messages on or off)” on page 954
- “SET TOCLOADRETENTION (Set load retention period for table of contents)” on page 955

## SET ACCOUNTING (Set accounting records on or off)

Use this 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.

Use the QUERY STATUS command to determine whether accounting records are generated. At installation, this value is set to OFF.

The accounting records are stored in an accounting file named `dsmacct.log`.

The environment variable, `DSMSERV_ACCOUNTING_DIR`, specifies the directory where the accounting file is located.

For more information about accounting, see the *Administrator's Guide*.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>> Set Accounting [ON | OFF]
  
```

### 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.

### Example: Create accounting records

To create an accounting record at the end of each client node session issue the command:

```
set accounting on
```

### Related commands

Table 290. Commands related to SET ACCOUNTING

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET ACTLOGRETENTION (Set the retention period or the size of the activity log)

Use this command to manage the activity log records by date or size. The activity log contains 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.

Activity log information includes messages, such as the following:

- Client session starts and ends
- Migration starts and ends
- Diagnostic error messages
- Scheduled administrative command output

You can choose to adjust the length of time that the activity log retains messages to avoid insufficient or outdated data. The server automatically removes the messages from the activity log after the retention period passes.

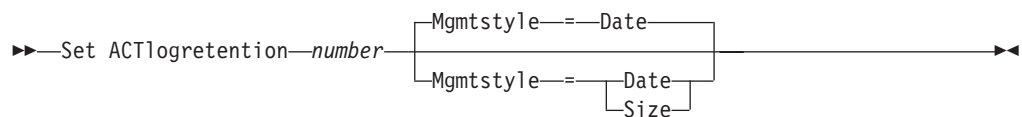
Alternatively, you can choose to limit the total size of the activity log to control the amount of space occupied by the activity log. The server will periodically remove the oldest activity log records until the activity log size no longer exceeds the configured maximum size allowed.

You can issue the QUERY STATUS command to display the current number of records in the activity log and the size (in megabytes) that the activity log currently occupies.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *number* (Required)

Specifies the number of days to retain messages in the activity log when the log is managed by date, or specifies the maximum size of the activity log when it is managed by size. With retention-based management, a value of 1 specifies to retain the activity log records only for the current day. With size-based management, a value of 1 specifies a maximum size of 1 MB for the activity log. You can specify a number from 0 to 9999. A value of 0 disables activity log retention.

#### **Mgmtstyle**

Specifies whether activity log management is retention-based or size-based. This parameter is optional. The default is DATE. Possible values are:

##### **Date**

Specifies that activity log management is retention-based.

## SET ACTLOGRETENTION

### Size

Specifies that activity log management is size-based.

### Example: Set the activity log retention period

Set the server to retain activity log records for 30 days. Issue the command:

```
set actlogretention 30
```

### Example: Set the activity log size

Set the server to limit the size of the activity log to 300 MB. Issue the command:

```
set actlogretention 300 mgmtstyle=size
```

### Related commands

*Table 291. Command related to SET ACTLOGRETENTION*

| Command      | Description   |
|--------------|---|
| QUERY ACTLOG | Displays messages from the server activity log.   |
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET ARCHIVERETENTIONPROTECTION (Activate data retention protection)

Use this command to activate and deactivate archive data retention protection. The server cannot contain any data in order for this command to work. At installation, the value is set to OFF.

When archive data retention protection is active:

- Only archive copies can be stored on the server.
- No archive copy can be deleted until the RETVER parameter in the DEFINE COPYGROUP (archive) command has been satisfied.

Defining storage pools of type RECLAMATIONTYPE=SNAPLOCK is only supported on servers with data retention protection enabled.

Use the QUERY STATUS command to display the status of archive data retention protection.

### Privilege class

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

### Syntax

```

>>—Set ARCHIVERETENTIONPROTECTION—Off—
                                   |
                                   ON

```

### Parameters

#### OFF

Specifies that archive data retention protection is not active.

#### ON

Specifies the archive data retention protection is active.

### Example: Activate data retention protection

Activate archive data retention protection by issuing the following command:

```
set archiveretentionprotection on
```

### Related commands

Table 292. Commands related to SET ARCHIVERETENTIONPROTECTION

| Command            | Description   |
|--------------------|---|
| ACTIVATE POLICYSET | Validates and activates a policy set.   |
| AUDIT VOLUME       | Compares database and storage pool information, and optionally, resolves any inconsistencies. |
| DEFINE COPYGROUP   | Defines a copy group for backup or archive processing within a specified management class.    |
| DEFINE VOLUME      | Assigns a volume to be used for storage within a specified storage pool.                      |

## SET ARCHIVERETENTIONPROTECTION

*Table 292. Commands related to SET ARCHIVERETENTIONPROTECTION (continued)*

| Command          | Description   |
|------------------|---|
| DELETE FILESPACE | Deletes data associated with client's file spaces.                                      |
| QUERY COPYGROUP  | Displays the attributes of a copy group.  |
| QUERY STATUS     | Displays the settings of server parameters, such as those selected by the SET commands. |
| UPDATE COPYGROUP | Changes one or more attributes of a copy group.   |

## SET AUTHENTICATION (Set password authentication)

Use this command to specify whether administrators and client nodes need a password to access the server. The value is set to ON at installation.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set Authentication—ON  
Off—►►
```

### 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.

### Example: Set authentication on

Set authentication to ON to require administrators and client nodes to enter a password when accessing the server.

```
set authentication on
```

### Related commands

*Table 293. Command related to SET AUTHENTICATION*

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET CLIENTACTDURATION (Set the duration period for the client action)

Use this command to specify the duration for the schedule that was defined with the DEFINE CLIENTACTION command. A client action defines a schedule that runs one time on a client.

The program deletes these event records whether or not the client has processed the schedule. However, the schedules are not deleted until after the first event records are deleted. The retention period for events defaults to 10 days at installation.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—SET CLIENTACTDuration—*days*—————►►

### Parameters

#### *days* (Required)

Specifies the number of days during which the schedule for the client action is active. You can specify an integer from 0 to 999. The default is 5 days.

The number of days you specify determines how long the database retains the schedule before deletion. A value of 0 indicates that the schedule duration is indefinite, and the schedule and associations are not deleted from the database.

### Example: Set a 15-day duration period for the client action

To specify that the schedule for the client action be active for 15 days issue the following command.

```
set clientactduration 15
```

### Related commands

Table 294. Commands related to SET CLIENTACTDURATION

| Command             | Description   |
|---------------------|---|
| DEFINE CLIENTACTION | Defines a command to be performed at a client node.                                     |
| QUERY STATUS        | Displays the settings of server parameters, such as those selected by the SET commands. |



## SET CONFIGMANAGER (Specify a configuration manager)

Use this command to specify whether a server is a configuration manager. On a configuration manager, you can define configuration profiles to which other servers can subscribe.

You cannot designate a server as a configuration manager if the server subscribes to one or more profiles on another configuration manager.

If a server is a configuration manager, you cannot change this designation until you delete all profiles, including the default profile.

Issue the QUERY STATUS command to determine if a server is a configuration manager. When a server is installed, it is not designated as a configuration manager.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### ON

Specifies that the server is a configuration manager.

When you designate a server as a configuration manager, IBM Tivoli Storage Manager creates a default profile named DEFAULT\_PROFILE and associates with the profile all servers and server groups defined on the configuration manager. You can modify or delete the default profile.

#### Off

Specifies that the server is not a configuration manager.

### Example: Specify a configuration manager

Designate a server as a configuration manager.

```
set configmanager on
```

### Related commands

Table 295. Commands related to SET CONFIGMANAGER

| Command           | Description   |
|-------------------|---|
| DEFINE PROFILE    | Defines a profile for distributing information to managed servers.                      |
| QUERY STATUS      | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET CONFIGREFRESH | Specifies a time interval for managed servers to contact configuration managers.        |

## SET CONFIGREFRESH (Set managed server configuration refresh)

Use this command on a managed server to specify how often that server contacts its configuration manager for updated configuration information.

To display the current setting, issue the QUERY STATUS command. At installation, the interval is set to 60 minutes.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set CONFIGRefresh—*minutes*—————►►

### Parameters

#### *minutes* (Required)

Specifies the interval, in minutes, before a managed server contacts its configuration manager for configuration updates. Specify an integer from 0 to 10000.

- If the value is greater than 0, the managed server immediately contacts the configuration manager. The next contact occurs when the specified interval is reached.
- If the value is 0, the managed server does not contact the configuration manager.

This value is ignored if the server does not subscribe to at least one profile on a configuration manager.

### Example: Set a 45-minute refresh interval

Specify that a managed server contacts its configuration manager every 45 minutes.

```
set configrefresh 45
```

### Related commands

Table 296. Commands related to SET CONFIGREFRESH

| Command                | Description  |
|------------------------|--|
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                 |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| NOTIFY SUBSCRIBERS     | Notifies servers to refresh their configuration information.       |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |
| UPDATE PROFILE         | Changes the description of a profile.                              |

# SET CONTEXTMESSAGING (Set message context reporting on or off)

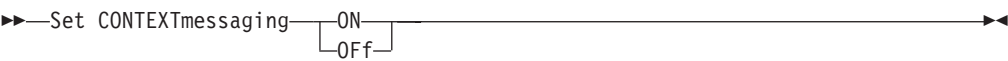
Use this command to get additional information when ANR9999D messages occur. Tivoli Storage Manager polls the server components for information that includes process name, thread name, session ID, transaction data, locks that are held, and database tables that are in use.

**Note:** When consecutive messages are issued from the same code area by the same thread, only the first of these messages will report the context information.

## Privilege class

To issue this command, you must have system privilege.

## Syntax



## Parameters

- ON**  
Specifies to enable message context reporting.
- OFF**  
Specifies to disable message context reporting.

## Example: Set message context reporting on or off

Turn on context messaging to receive additional information that could help determine the cause of ANR9999D messages.

```
set contextmessaging on
```

## Related commands

Table 297. Commands related to SET CONTEXTMESSAGING

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET CROSSDEFINE (Specifies whether to cross-define servers)

Use this command to specify whether a server is automatically defined to another server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>> Set CROSSDefine ON  
Off

```

### Parameters

#### ON

Specifies that a server may be cross-defined to another server. To automatically define one server to another, you must also permit cross defining in the server definition.

#### OFF

Specifies that a server may not be cross-defined to another server.

### Example: Specifies whether to cross-define servers

Set cross define on to allow a server to be cross-defined to another server.  
 set crossdefine on

### Related commands

*Table 298. Command related to SET CROSSDEFINE*

| Command             | Description   |
|---------------------|---|
| DEFINE SERVER       | Defines a server for server-to-server communications. |
| SET SERVERHLADDRESS | Specifies the high-level address of a server.         |
| SET SERVERLLADDRESS | Specifies the low-level address of a server.          |
| SET SERVERPASSWORD  | Specifies the server password.                        |

## SET DBRECOVERY (Set the device class for automatic backups)

Use this command to specify the device class to be used for automatic backups.

If you issue the BACKUP DB command, and the device class is not the one that is specified in the SET DBRECOVERY command, a warning message is issued. However, the backup operation continues and is not affected.

### Privilege class

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

### Syntax

►►—SET DBRECOV~~er~~y—*device\_class\_name*—◄◄

### Parameters

*device\_class\_name* **(Required)**

Specifies the device class to use for database backups.

### Example: Specify a device class for database backups

Specify the DBBACK device class for database backups. Issue the command:

```
set dbrecovery dbback
```

### Related commands

Table 299. Commands related to SET DBRECOVERY

| Command       | Description  |
|---------------|--|
| BACKUP DB     | Backs up the IBM Tivoli Storage Manager database to sequential access volumes. |
| QUERY DB      | Displays allocation information about the database.                            |
| QUERY DBSPACE | Displays information about the storage space defined for the database.         |

## SET DBREPORTMODE (Set the level of database reporting)

Use this command to set the level of database diagnostic reporting to be presented on the service console and activity log.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### None

Specifies that no database diagnostic reporting is performed. This setting should not normally be used.

#### Partial

Specifies that the only events that are reported are those that are exceptions and that might represent errors. This is the default.

#### Full

Specifies that all database diagnostic information is reported. This includes information that is normal and expected.

### Example: Report only events that are likely to be errors

Issue the following command:

```
set dbreportnode partial
```

### Related commands

Table 300. Commands related to SET DBREPORTMODE

| Command      | Description                                     |
|--------------|---|
| QUERY ACTLOG | Displays messages from the server activity log. |

## SET DRMACTIVEDATASTGPOOL (Specify the active-data pools to be managed by DRM)

Use this command to specify names of the active-data pools to be recovered after a disaster. Tivoli Storage Manager uses these names if the PREPARE , MOVE DRMEDIA, or QUERY DRMEDIA command does not include the ACTIVATEDATASTGPOOL parameter.

By default, volumes in active-data pools are not eligible for processing by disaster recovery manager. To process active-data pool volumes, you must issue the SET DRMACTIVEDATASTGPOOL command, or you must use the ACTIVATEDATASTGPOOL command-line parameter on the MOVE DRMEDIA, QUERY DRMEDIA, or PREPARE command.

Use the QUERY DRMSTATUS command to display the current settings.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>>> Set DRMACTIVEDatastgpool active-data_pool_name

```

### Parameters

#### *active-data\_pool\_name* (Required)

Specifies the active-data pool names. Separate multiple names with commas with no intervening spaces. You can use wildcard characters. The specified names will overwrite any previous settings. If you enter a null string (""), all current names are removed, and no active-data pool volumes in MOUNTABLE state are processed if they were not explicitly entered as MOVE DRMEDIA , QUERY DRMEDIA, or PREPARE command parameters.

### Example: Set an eligible active-data pool

Set ACTIVEDATAPOOL1 as the eligible active-data pool.

```
set drmactivedatapool activedatastgpool1
```

### Related commands

Table 301. Commands related to SET DRMACTIVEDATASTGPOOL

| Command            | Description   |
|--------------------|---|
| MOVE DRMEDIA       | Moves DRM media on-site and off-site.                 |
| PREPARE            | Creates a recovery plan file.                         |
| QUERY DRMEDIA      | Displays information about disaster recovery volumes. |
| QUERY DRMSTATUS    | Displays DRM system parameters.                       |
| SET DRMCOPYSTGPOOL | Specifies that copy storage pools are managed by DRM. |

## SET DRMACTIVEDATASTGPOOL

*Table 301. Commands related to SET DRMACTIVEDATASTGPOOL (continued)*

| Command            | Description  |
|--------------------|--|
| SET DRMPRIMSTGPOOL | Specifies that primary storage pools are managed by DRM. |



SET DRMCHECKLABEL (Specify label checking)

Use this command to specify whether Tivoli Storage Manager reads the labels of sequential media checked out by the MOVE DRMEDIA command. At installation, the value of the DRMCHECKLABEL is set to YES.

Use the QUERY DRMSTATUS command to check the current setting.

This command does not apply to 349X device types.

Privilege class

To issue this command, you must have system privilege.

Syntax



Parameters

- Yes**  
Specifies that Tivoli Storage Manager reads the labels of sequential media checked out by the MOVE DRMEDIA command.
- No**  
Specifies that Tivoli Storage Manager does not read the labels of sequential media checked out by the MOVE DRMEDIA command.

Example: Specify no label checking

Specify that Tivoli Storage Manager does not perform label checking.  
`set drmchecklabel no`

Related commands

Table 302. Commands related to SET DRMCHECKLABEL

| Command         | Description                           |
|-----------------|---------------------------------------|
| MOVE DRMEDIA    | Moves DRM media on-site and off-site. |
| QUERY DRMSTATUS | Displays DRM system parameters.       |

## SET DRMCMDFILENAME (Specify the name of a file to contain commands)

Use this command to name a file that can contain the commands created when the MOVE DRMEDIA or QUERY DRMEDIA commands are issued. If the SET DRMCMDFILENAME is not issued, the MOVE DRMEDIA or QUERY DRMEDIA command generates a file name.

Use the QUERY DRMSTATUS command to display the current command file name.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMCMDFilename—*file\_name*—◄◄

### Parameters

#### *file\_name* (Required)

Specifies a full path name for a file to contain the commands created by the MOVE DRMEDIA or QUERY DRMEDIA command.

**Attention:** If a file of the same name already exists, MOVE DRMEDIA or QUERY DRMEDIA command tries to use it, and the existing data is overwritten.

### Example: Specify a file name to contain DRMEDIA commands

Specify a file name of /adsm/drm/orm/exec.cmds.

```
set drmcmdfilename /adsm/drm/orm/exec.cmds
```

### Related commands

Table 303. Commands related to SET DRMCMDFILENAME

| Command         | Description   |
|-----------------|---|
| MOVE DRMEDIA    | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA   | Displays information about disaster recovery volumes. |
| QUERY DRMSTATUS | Displays DRM system parameters.                       |

## SET DRMCOPYSTGPOOL (Specify the copy storage pools to be managed by DRM)

Use this command to specify names of the copy storage pools to be recovered after a disaster. Tivoli Storage Manager uses these names if the PREPARE command does not include the COPYSTGPOOL parameter.

If the MOVE DRMEDIA or QUERY DRMEDIA command does not include the COPYSTGPOOL parameter, the command processes the volumes in the MOUNTABLE state that are in the copy storage pool named by the SET DRMCOPYSTGPOOL command. At installation, all copy storage pools are eligible for DRM processing.

Use the QUERY DRMSTATUS command to display the current settings.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶—Set DRMCOPYstgpool—copy_pool_name—▶▶

```

### Parameters

#### *copy\_pool\_name* (Required)

Specifies the copy storage pool names. Separate multiple names with commas and no intervening spaces. You can use wildcard characters. The specified names replace any previous setting. If you enter a null string (""), all current names are removed, and all copy storage pools are eligible for processing.

### Example: Set an eligible copy storage pool

Set COPYSTGPOOL1 as the eligible copy storage pool.

```
set drmcopystgpool copystgpool1
```

### Related commands

Table 304. Commands related to SET DRMCOPYSTGPOOL

| Command            | Description  |
|--------------------|--|
| MOVE DRMEDIA       | Moves DRM media on-site and off-site.                    |
| PREPARE            | Creates a recovery plan file.                            |
| QUERY DRMEDIA      | Displays information about disaster recovery volumes.    |
| QUERY DRMSTATUS    | Displays DRM system parameters.                          |
| SET DRMPRIMSTGPOOL | Specifies that primary storage pools are managed by DRM. |

## SET DRMCOURIERNAME (Specify the courier name)

Use this command to specify the courier name. At installation, this name is set to COURIER. The MOVE DRMEDIA command uses the courier name to set the location of volumes that are moving to the COURIER state.

You can use the QUERY DRMSTATUS to see the name of the courier.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMCOURiername—*courier\_name*—◄◄

### Parameters

#### *courier\_name* (Required)

Specifies the name of the courier. The name can be up to 255 characters. Enclose the name in quotation marks if it contains any blank characters.

### Example: Set the courier name

Set the name of the courier to Joe's Courier Service.

```
set drmcouriername "Joe's Courier Service"
```

### Related commands

Table 305. Commands related to SET DRMCOURIERNAME

| Command         | Description   |
|-----------------|---|
| MOVE DRMEDIA    | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA   | Displays information about disaster recovery volumes. |
| QUERY DRMSTATUS | Displays DRM system parameters.                       |

## SET DRMDBBACKUPEXPIREDAYS (Specify DB backup series expiration)

Use this command to specify when a database backup series is eligible to be expired.

The value set by this command applies to both a snapshot and a full plus incremental database backup series. Any type of database backup series is eligible for expiration if all of the following are true:

- The age of the last volume of the series has exceeded the expiration value set by this command.
- For volumes that are not virtual volumes, all volumes in the series are in the VAULT state.
- The volume is not part of the most recent database backup series.

**Remember:** The most recent backup series of either type is not deleted. See the MOVE DRMEDIA command for more information on the expiration of database backup volumes that are not virtual volumes. See the EXPIRE INVENTORY command for more information on expiration of database backup volumes that are virtual volumes.

Use the QUERY DRMSTATUS to see the number of days specified.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMDBBackupexpiredays—days—————►►

### Parameters

#### *days* (Required)

Specifies the number of days that must elapse since a database series was created before it is eligible to be expired. The number of days must match the volume reuse delay period for copy storage pools managed by disaster recovery manager. Specify an integer from 0 to 9999.

### Example: Set the database backup series expiration

Set the database backup series expiration value to 60.

```
set drmdbbackupexpiredays 60
```

### Related commands

Table 306. Commands related to SET DRMDBBACKUPEXPIREDAYS

| Command            | Description   |
|--------------------|---|
| DSMSERV RESTORE DB | Restores an IBM Tivoli Storage Manager database.      |
| MOVE DRMEDIA       | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA      | Displays information about disaster recovery volumes. |

## SET DRMDBBACKUPEXPIREDAYS

*Table 306. Commands related to SET DRMDBBACKUPEXPIREDAYS (continued)*

| Command          | Description   |
|------------------|---|
| QUERY DRMSTATUS  | Displays DRM system parameters.   |
| QUERY VOLHISTORY | Displays sequential volume history information that has been collected by the server. |

## SET DRMFILEPROCESS (Specify file processing)

Use this command to specify if the MOVE DRMEDIA or QUERY DRMEDIA command should process database backup volumes and copy storage pool volumes that are associated with a FILE device class. At installation, the value is set to NO. Use the QUERY DRMSTATUS to determine the current setting.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### No

Specifies that the MOVE DRMEDIA and QUERY DRMEDIA commands does not process database backup and copy storage pool volumes that are associated with a FILE device class. This is the default.

#### Yes

Specifies that the MOVE DRMEDIA and QUERY DRMEDIA commands process database backup and copy storage pool volumes that are associated with a FILE device class.

### Example: Specify that the DRMEDIA commands do not include FILE type device classes

Set the file processing value to no.

```
set drmfileprocess no
```

### Related commands

Table 307. Commands related to SET DRMFILEPROCESS

| Command         | Description   |
|-----------------|---|
| MOVE DRMEDIA    | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA   | Displays information about disaster recovery volumes. |
| QUERY DRMSTATUS | Displays DRM system parameters.                       |

## SET DRMINSTRPREFIX (Specify the prefix for recovery instructions file names)

Use this command to specify a prefix to the recovery instructions file name. If you issue this command, IBM Tivoli Storage Manager uses the specified prefix if the PREPARE command is issued without the INSTRPREFIX parameter.

Use the QUERY DRMSTATUS command to display the current value for the prefix.

the prefix is the current Tivoli Storage Manager server working directory.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMINSTRPrefix—*prefix*—————►►

### Parameters

#### *prefix* (Required)

Specifies a path name prefix for the files that contain the recovery instructions. When processing the PREPARE command, Tivoli Storage Manager appends the name of the appropriate recovery plan file stanza to find the file. The maximum length is 250 characters.

The prefix can be one of the following:

- **Directory path:** End the prefix with a forward slash (/). For example:  
/admsrv/recinstr/

For the RECOVERY.INSTRUCTIONS.GENERAL file, the resulting file name would be:

/admsrv/recinstr/RECOVERY.INSTRUCTIONS.GENERAL

- **Directory path followed by a string:** Tivoli Storage Manager treats the string as part of the file name. For example:  
/admsrv/recinstr/accounts

For the RECOVERY.INSTRUCTIONS.GENERAL file, the resulting file name would be:

/admsrv/recinstr/accounts.RECOVERY.INSTRUCTIONS.GENERAL

- **String only:** Tivoli Storage Manager specifies the directory path and appends the appropriate recovery plan file stanza name.
  - Tivoli Storage Manager uses the name of the current working directory. For example, the current working directory is /opt/tivoli/tsm/server/bin. You specify the following:  
shipping

For the RECOVERY.INSTRUCTIONS.GENERAL file, the resulting file name would look like this:

/opt/tivoli/tsm/server/bin/shipping.RECOVERY.INSTRUCTIONS.GENERAL



**Example: Specify the recovery plan prefix**

Specify reading the recovery plan instructions from directory `/drmpplan/primesrv`.  
`set drminstrprefix /drmpplan/primesrv/`

**Related commands**

*Table 308. Commands related to SET DRMINSTRPREFIX*

| Command         | Description                     |
|-----------------|---------------------------------|
| PREPARE         | Creates a recovery plan file.   |
| QUERY DRMSTATUS | Displays DRM system parameters. |

## SET DRMNOTMOUNTABLENAME (Specify the not mountable location name)

Use this command to specify the name of the onsite location for storing the media. At installation, the name is set to NOTMOUNTABLE. Use the QUERY DRMSTATUS command to see the location name.

The location name is used by the MOVE DRMEDIA command to set the location of volumes that are moving to the NOTMOUNTABLE state.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set DRMNOTMountableName—location—————►◄
```

### Parameters

#### *location* (Required)

Specifies the name of the onsite location for storing the media. The name can be up to 255 characters. Enclose the name in quotation marks if it contains any blank characters.

### Example: Specify the name of the onsite location

Set the name of the location to room123/31.

```
set drmnountable "room 123/31"
```

### Related commands

*Table 309. Commands related to SET DRMNOTMOUNTABLENAME*

| Command         | Description   |
|-----------------|---|
| MOVE DRMEDIA    | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA   | Displays information about disaster recovery volumes. |
| QUERY DRMSTATUS | Displays DRM system parameters.                       |

## SET DRMPLANPREFIX (Specify a prefix for recovery plan file names)

Use this command to specify a prefix for a recovery plan file name.

If you issue this command, Tivoli Storage Manager uses the specified prefix if the PREPARE command does not include the PLANPREFIX parameter.

Use the QUERY DRMSTATUS command to display the current value for the recovery plan prefix.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMPLANPrefix—*prefix*—————►►

### Parameters

#### *prefix* (Required)

Specifies the prefix for a recovery plan file name. The maximum length of the prefix is 250 characters. If you enter a null string (""), the current prefix is removed, and the server uses the algorithm described in the PLANPREFIX parameter in the PREPARE command.

For the prefix, you can specify:

- **A directory path followed by a forward slash (/):** Tivoli Storage Manager appends to the prefix the date and time in the `yyyymmdd.hhmmss` format. For example, the SET DRMPLANPREFIX is set to the following:

```
/admsrv/recplans/
```

The resulting recovery plan file name is:

```
/admsrv/recplans/19971115.051421
```

- **A directory path followed by a string:** Tivoli Storage Manager uses the string as part of the file name. Tivoli Storage Manager appends to the prefix the date and time in the `.yyyymmdd.hhmmss` format (note the initial period). For example, the SET DRMPLANPREFIX is set to the following:

```
/admsrv/recplans/accounting
```

The resulting recovery plan filename is:

```
/admsrv/recplans/accounting.19971115.051421
```

- **A string that is not preceded by a directory path:** Tivoli Storage Manager appends to the prefix the date and time information in the `.yyyymmdd.hhmmss` format (note the initial period). Tivoli Storage Manager determines the directory path as follows:
  - Tivoli Storage Manager uses the directory path name of the current working directory of the Tivoli Storage Manager server. For example, the current Tivoli Storage Manager working directory is `/opt/tivoli/tsm/server/bin`. The SET DRMPLANPREFIX command is set to the following: `shipping`

The resulting recovery plan file name is:

## SET DRMPLANPREFIX

/opt/tivoli/tsm/server/bin/shipping.19971115.051421

### Example: Specify a prefix for recovery plan file names

Specify a prefix so that the generated recovery plan files are stored in the following directory:

/drmpln/primsrv

Issue the command:

```
set drmplnprefix /drmpln/primsrv/
```

### Related commands

*Table 310. Commands related to SET DRMPLANPREFIX*

| Command         | Description                     |
|-----------------|---------------------------------|
| PREPARE         | Creates a recovery plan file.   |
| QUERY DRMSTATUS | Displays DRM system parameters. |

## SET DRMPLANVPOSTFIX (Specify replacement volume names)

Use this command to specify the character to be appended to replacement volume names in the recovery plan file. The character can help you find or generate replacement volume names when you use the recovery plan file.

At installation, the character is set to @. Tivoli Storage Manager generates replacement names for primary storage pool volumes that were added by the DEFINE VOLUME command. Use the appended character to:

- Find replacement volume names in the recovery plan stanzas so that you can change the names at recovery time. For example, you may not know the names of the available tape volumes at the recovery site.
- Generate replacement volume names. You need a naming convention that works for any device type in your primary storage pools. Consider the following:
  - The generated length of replacement volume name
  - Legal characters in the replacement volume name
  - Conflicts with existing volume names
  - A replacement volume name must be different from any destroyed, existing, or new volume name.

Use the QUERY DRMSTATUS command to see the character added to the end of the replacement volume names.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMPLANVpostfix—*character*—————►►

### Parameters

#### *character* (Required)

Specifies the character appended to the replacement volume names in the recovery plan file. Specify an alphanumeric or special character.

**Attention:** A special character can cause unpredictable results in the AIX shell or command line environment.

### Example: Specify the appended character for replacement volume names

Set the character appended to the replace volume names to R.

```
set drmplnvpostfix R
```

### Related commands

Table 311. Commands related to SET DRMPLANVPOSTFIX

| Command         | Description                     |
|-----------------|---------------------------------|
| PREPARE         | Creates a recovery plan file.   |
| QUERY DRMSTATUS | Displays DRM system parameters. |

## SET DRMPRIMSTGPOOL (Specify the primary storage pools to be managed by DRM)

Use this command to specify the names of primary storage pools that you want to recover. If the PREPARE command does not include the PRIMSTGPOOL parameter, DRM processes the names specified in this command.

Use the QUERY DRMSTATUS command to display the current settings. At installation, all primary storage pools defined to the server are eligible for DRM processing.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>>—Set DRMPRIMstgpool—primary_pool_name—>>

```

### Parameters

#### *primary\_pool\_name* (Required)

Specifies the names of the primary storage pool names you want to recover. Separate multiple names with commas and no intervening spaces. You can use wildcard characters to specify names. The names that you specify replace any previous setting. If you enter a null string (""), all current names are removed, and all primary storage pools are eligible for DRM processing.

### Example: Set a primary storage pool to be managed by DRM

Set the primary storage pool to be managed by DRM to PRIMSTGPOOL1.

```
set drmprimstgpool primstgpool1
```

### Related commands

Table 312. Commands related to SET DRMPRIMSTGPOOL

| Command            | Description   |
|--------------------|---|
| PREPARE            | Creates a recovery plan file.                         |
| QUERY DRMSTATUS    | Displays DRM system parameters.                       |
| SET DRMCOPYSTGPOOL | Specifies that copy storage pools are managed by DRM. |

## SET DRMRPFEXPIREDAYS (Set criteria for recovery plan file expiration)

Use this command to specify when recovery plan files are eligible for expiration. This command and expiration processing apply only to recovery plan files that were created with the DEVCLASS parameter specified on the PREPARE command (that is, virtual volumes of type RPFIL and RPSNAPSHOT). Expiration processing on the source server expires plan files stored on the target server. Locally created recovery plan files are not expired.

An RPFIL file is associated with a full plus incremental database backup series. An RPSNAPSHOT file is associated with a database snapshot backup series.

**Attention:** The latest RPFIL and RPSNAPSHOT files are never deleted.

A recovery plan file is eligible for expiration if both of the following are true:

- The last recovery plan file of the series has exceeded the expiration value specified with the SET DRMRPFEXPIREDAYS command.
- The latest recovery plan file is not associated with the most recent database backup series.

See the EXPIRE INVENTORY command for more information about expiration processing.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DRMRPFExpiredays—days—————►►

### Parameters

#### *days* (Required)

Specifies the number of days that must elapse before a recovery plan file expires. You can specify a number from 0 to 9999. At installation, this value is set to 60.

### Example: Set the recovery plan expiration

Set the recovery plan file expiration value to 30.

```
set drmrpfexpiredays 30
```

### Related commands

*Table 313. Commands related to SET DRMRPFEXPIREDAYS*

| Command          | Description                                     |
|------------------|---|
| PREPARE          | Creates a recovery plan file.                   |
| QUERY DRMSTATUS  | Displays DRM system parameters.                 |
| QUERY RPFCONTENT | Displays the contents of a recovery plan file.  |
| QUERY RPFIL      | Displays information about recovery plan files. |

## SET DRMRPFEXPIREDAYS

*Table 313. Commands related to SET DRMRPFEXPIREDAYS (continued)*

| Command                   | Description   |
|---------------------------|---|
| QUERY VOLHISTORY          | Displays sequential volume history information that has been collected by the server. |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.                             |



## SET DRMVAULTNAME (Specify the vault name)

Use this command to specify the vault name. At installation the name is set to VAULT. Use the QUERY DRMSTATUS command to see the name of the vault.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—SET DRMAu1tname—*vault\_name*—◄◄

### Parameters

#### *vault\_name* (Required)

Specifies the name of the vault. The name can be up to 255 characters. Enclose the name in quotation marks if it contains any blank characters.

### Example: Specify a vault name

Specify ironmountain as the vault name.

```
set drmvaultname ironmountain
```

### Related commands

Table 314. Commands related to SET DRMVAULTNAME

| Command         | Description   |
|-----------------|---|
| MOVE DRMEDIA    | Moves DRM media on-site and off-site.                 |
| QUERY DRMEDIA   | Displays information about disaster recovery volumes. |
| QUERY DRMSTATUS | Displays DRM system parameters.                       |

## SET EVENTRETENTION (Set the retention period for event records)

Use this command to set the retention period for event records in the server database that will allow you to monitor completed schedules. An event record is created whenever processing of a scheduled command is started or missed.

You can adjust the length of time that the server maintains event information to avoid insufficient or outdated data. The server automatically removes the event records from the database after the retention period passes and the startup window for the event has elapsed.

You can issue the QUERY EVENT command to display information about scheduled and completed events.

You can issue the DELETE EVENT command to delete event records regardless of whether their retention period has passed.

You can issue the QUERY STATUS command to display the value for the event retention period. 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* (Required)

The number of days that the database retains event records. You can specify an integer from 0 to 9999. A value of 0 indicates that only event records for the current day are retained.

### Example: Set the retention period for event records

Set the retention period to 15 days.

```
set eventretention 15
```

### Related commands

*Table 315. Commands related to SET EVENTRETENTION*

| Command      | Description   |
|--------------|---|
| DELETE EVENT | Deletes event records prior to a specified date and time.                               |
| QUERY EVENT  | Displays information about scheduled and completed events for selected clients.         |
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET INVALIDPWLIMIT (Set the number of invalid logon attempts)

Use this command to set the number of invalid logon attempts allowed before a node is locked.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set—INVALIDPwlimit—*number*—►►

### Parameters

#### *number* (Required)

Specifies the number of invalid logon attempts allowed before a node is locked.

You can specify an integer from 0 to 9999. A value of 0 means that invalid logon attempts are not checked. A value of 1 means that if a user issues an invalid password once, the node is locked by the server. The default is 0.

### Example: Define the number of allowed invalid login attempts

Set the number of invalid logon attempts allowed.

```
set invalidpwlimit 6
```

### Related commands

Table 316. Commands related to SET INVALIDPWLIMIT

| Command         | Description   |
|-----------------|---|
| QUERY ADMIN     | Displays information about one or more IBM Tivoli Storage Manager administrators.       |
| QUERY NODE      | Displays partial or complete information about one or more clients.                     |
| QUERY STATUS    | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET MINPWLENGTH | Sets the minimum length for client passwords.   |

## SET LICENSEAUDITPERIOD (Set license audit period)

Use this command to specify the period, in days, between automatic license audits performed by IBM Tivoli Storage Manager.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

▶▶—Set—LICenseauditperiod—30—days—▶▶

```

### Parameters

*days*

Specifies the number of days between automatic server license audits. This parameter is optional. The default value is 30. You can specify an integer from 1 to 30, inclusive.

### Example: Specify a 14 day server license audit

Specify that the server audits licenses every 14 days.

```
set licenseauditperiod 14
```

### Related commands

*Table 317. Commands related to SET LICENSEAUDITPERIOD*

| Command              | Description   |
|----------------------|---|
| AUDIT LICENSES       | Checks for compliance with defined licenses.  |
| QUERY AUDITOCCUPANCY | Displays the server storage utilization for a client node.                              |
| QUERY LICENSE        | Displays information about licenses and audits.   |
| QUERY STATUS         | Displays the settings of server parameters, such as those selected by the SET commands. |
| REGISTER LICENSE     | Registers a new license with the IBM Tivoli Storage Manager server.                     |

## SET MAXCMDRETRIES (Set the maximum number of command retries)

Use this command to set the maximum number of times that a scheduler on a client node can retry a failed, scheduled command.

You can use the command to override the maximum number of retries that are specified by the client node. A client's value is overridden only if the client is able to connect with the server.

This command is used with the SET RETRYPERIOD command to regulate the time and the number of retry attempts to rerun failed command.

You can issue the QUERY STATUS command to display the current retry value. At installation, Tivoli Storage Manager is configured so that each client determines its own retry value.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
➤➤—Set MAXCMDRetries—┐
                        └─number─┘
```

### Parameters

*number*

Specifies the maximum number of times the scheduler on a client node can retry a failed scheduled command. This parameter is optional.

The default is that each client determines its own value for this parameter. You can specify an integer from 0 to 9999. See the appropriate client documentation for more information on setting the maximum command retries from the client.

### Example: Set the maximum number of command retries to 2

Retry, only twice, a failed attempt to process a scheduled command.

```
set maxcmdretries 2
```

### Related commands

Table 318. Command related to SET MAXCMDRETRIES

| Command         | Description   |
|-----------------|---|
| QUERY STATUS    | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET RETRYPERIOD | Specifies the time between retry attempts by the client scheduler.                      |

## SET MAXSCHEDSESSIONS (Set maximum scheduled sessions)

Use this command to set the number of sessions that are allowed to the server for processing scheduled operations. This command specifies the maximum number of scheduled sessions as a percentage of the total number of available server sessions.

Limiting the number of sessions ensures that sessions are available for unscheduled operations, such as backup or archive.

You can increase either the total number of sessions (with the MAXSESSIONS parameter) or the maximum percentage of scheduled sessions. Increasing the total number of sessions available can affect server performance. Increasing the maximum percentage of scheduled sessions can reduce the sessions available for unscheduled operations.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set MAXSchedsessions—*percent*—►►

### Parameters

#### *percent* (Required)

Specifies the percentage of total server sessions that can be used for scheduled operations. You can specify an integer from 0 to 100. The MAXSESSIONS parameter in the server options file determines the maximum number of total available server sessions.

If you set the maximum percentage of scheduled sessions to 0, no scheduled events can begin. If you set the maximum percentage of scheduled sessions to 100, the maximum number of scheduled sessions is the value of the MAXSESSIONS option.

### Example: Set a maximum of 20 sessions for scheduled activities

The MAXSESSIONS option has a value of 80. To permit no more than 20 sessions to be available for scheduled activity, set the percentage to 25.

```
set maxschedsessions 25
```

### Related commands

Table 319. Commands related to SET MAXSCHEDSESSIONS

| Command      | Description   |
|--------------|---|
| QUERY OPTION | Displays information about server options.  |
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET MINPWLENGTH (Set minimum password length)

Use this command to set the minimum length of a password.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set—MINPwlength—*length*—◄◄

### Parameters

#### *length* (Required)

Specifies the minimum length of a password. You can specify an integer from 0 to 64. A value of 0 means that the password length is not checked. At installation, the value for minimum password length is set to 0.

### Example: Set the minimum password length

Set the minimum password length to 5 characters.

```
set minpwlenth 5
```

### Related commands

Table 320. Commands related to SET MINPWLENGTH

| Command            | Description   |
|--------------------|---|
| QUERY STATUS       | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET INVALIDPWLIMIT | Sets the number of invalid logon attempts before a node is locked.                      |

## SET PASSEXP (Set password expiration date)

Use this command to set the expiration period for administrator and client node passwords. You can set a common password expiration period for all administrators and client node passwords or selectively set password expiration periods.

You can override this setting for one or more nodes by using the REGISTER NODE or UPDATE NODE command with the PASSEXP parameter.

**Attention:** If you do not specify the NODE or ADMIN parameters, *all* client node and administrator passwords will use the new password expiration period. If you selectively set a password expiration period for a client node or administrator that does not already have a set password expiration period, it is not modified if you later set a password expiration for all users. The NODE or ADMIN parameters must be specified to change the password expiration period for client nodes or administrators that have selectively set password expiration periods.

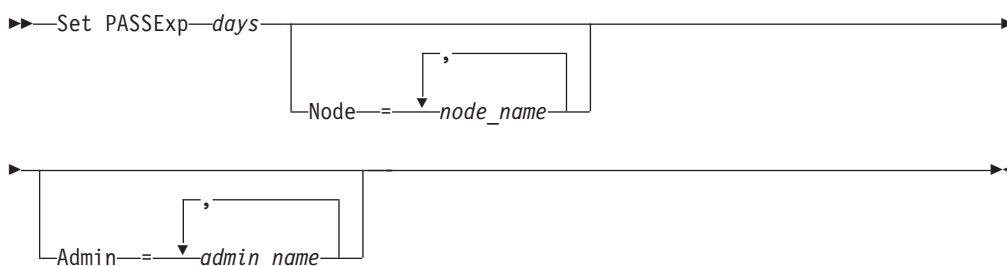
You can use the RESET PASSEXP command to reset the password expiration period to the common expiration period.

Use the QUERY STATUS command to display the common password expiration period, which at installation is set to 90 days.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *days* (Required)

Specifies the number of days that a password remains valid.

You can specify from 1 to 9999 if you do not specify the NODE or the ADMIN parameter. If you specify the NODE or the ADMIN parameter, you can specify from 0 to 9999. A value of 0 means that the password never expires. If a password expires, the server prompts for a new password when the administrator or client node contacts the server.

#### Node

Specifies the name of the node whose password expiration period you would like to set. To specify a list of nodes, separate the names with commas and no intervening spaces. This parameter is optional.

#### Admin

Specifies the name of the administrator whose password expiration period you



would like to set. To specify a list of administrators, separate the names with commas and no intervening spaces. This parameter is optional.

### Example: Set the administrator and client node password expiration

Set the administrator and client node password expiration period to 45 days.

```
set passexp 45
```

### Example: Set an administrator's password expiration

Set the administrator LARRY's password expiration period to 120 days.

```
set passexp 120 admin=larry
```

### Related commands

*Table 321. Commands related to SET PASSEXP*

| Command       | Description   |
|---------------|---|
| QUERY STATUS  | Displays the settings of server parameters, such as those selected by the SET commands. |
| REGISTER NODE | Defines a client to the server and sets options for that user.                          |
| RESET PASSEXP | Resets the password expiration for nodes or administrators.                             |
| UPDATE ADMIN  | Changes the password or contact information associated with any administrator.          |
| UPDATE NODE   | Changes the attributes associated with a client node.                                   |

## SET QUERYSCHEDPERIOD (Set query period for polling client nodes)

Use this command to regulate how often client nodes contact the server to obtain scheduled work when it is running in the client-polling scheduling mode.

Each client can set its own retry period at the time its scheduler is started. You can use this command to override the value specified by all clients that can connect with the server.

If client nodes poll more frequently for schedules, the nodes receive changes to schedules more quickly. However, increased polling by the client nodes also increases network traffic.

You can issue the QUERY STATUS command to display the value for the period between schedule queries. At installation, Tivoli Storage Manager 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—┐
                           |hours|
```

### Parameters

*hours*

Specifies the maximum number of hours the scheduler on a client node waits between attempts to contact the server to obtain a schedule. This parameter is optional. You can specify an integer from 1 to 9999. If you do not specify a value for this parameter, each client determines its own value for this parameter.

### Example: Set the polling period for all client nodes

Have all clients using the polling scheduling mode contact the server every 24 hours.

```
set queryschedperiod 24
```

### Related commands

Table 322. Commands related to SET QUERYSCHEDPERIOD

| Command        | Description   |
|----------------|---|
| QUERY STATUS   | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET SCHEDMODES | Specifies the central scheduling mode for the server.                                   |

## SET RANDOMIZE (Set randomization of scheduled start times)

Use this command to set randomized start times within the startup window of each schedule for clients by using the client-polling scheduling mode. A startup window is the start time and duration during which a schedule must be initiated. A client-polling scheduling mode is a client/server communication technique where the client queries the server for work.

Each schedule has a window during which it can be run. To balance network and server load, the start times for clients can be scattered across that window. Use this command to specify the fraction of the window over which start times for clients are distributed.

The randomization occurs at the beginning of the window to allow time for retries, if necessary. Randomization does not occur if the client's first contact with the server is after the start time for the event.

You can issue the QUERY STATUS command to display the value for the schedule randomization percentage. At installation, the value is 25 percent.

Set the randomization percentage to a value greater than 0 to prevent communication errors. Communication errors 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. This decreases the chance for communication overload and failure.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set RANDomize—*percent*—————►►

### Parameters

#### *percent* (Required)

Specifies the percentage of the startup window over which the start times for individual clients are distributed. You can specify an integer from 0 to 50.

A value of 0 indicates that no randomization occurs and that all clients run schedules at the beginning of the startup windows.

A value of 50 indicates that clients are assigned start times that are randomly scattered across the first half of each startup window.

At installation, this value is 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 is applied to a 24 hour period. For example, a value of 25 percent would result in a 6 hour window.

### Example: Set randomization of scheduled start times

Set randomization to 50 percent.

```
set randomize 50
```

### Related commands

*Table 323. Commands related to SET RANDOMIZE*

| Command         | Description   |
|-----------------|---|
| DEFINE SCHEDULE | Defines a schedule for a client operation or an administrative command.                 |
| QUERY STATUS    | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET SCHEDMODES  | Specifies the central scheduling mode for the server.                                   |

## SET REGISTRATION (Set open or closed registration)

Use this command to require a system or policy administrator to register client nodes (closed registration) or to permit each user to register his own workstation as a client node with the server (open registration).

With *closed* registration, an administrator defines the following:

- The node name and password for each workstation
- 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 can delete backup or archive files from server storage

With *open* registration, when a user accesses 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

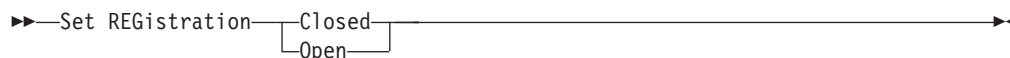
Existing registered client nodes are not affected by changes in the registration process.

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 by a system or policy administrator.

#### Open

Specifies that users can register their workstations as client nodes with the server.

## SET REGISTRATION

### Example: Set the policy so only administrators can register client nodes

Issue the following command to limit client node registration to a policy or system administrator.

```
set registration closed
```

### Example: Set the policy so users can register client nodes

Issue the following command to allow users register as client nodes to the server.

```
set registration open
```

### Related commands

*Table 324. Command related to SET REGISTRATION*

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET RETRYPERIOD (Set time between retry attempts)

Use this command to set the number of minutes the scheduler on a client node waits between retry attempts after a failed attempt to contact the server or after a scheduled command fails to process.

Each client can set its own retry period at the time its scheduler program is started. You can use this command to override the values specified by all clients that can connect with the server.

This command is used in conjunction with the SET MAXCMDRETRIES command to regulate the period of time and the number of retry attempts to run a failed command.

You can issue the QUERY STATUS command to display the value for the period between retries. At installation, Tivoli Storage Manager allows each client to determine its own retry period.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
➤➤—Set RETRYPeriod—┐
                      └minutes┘➤➤
```

### Parameters

#### *minutes*

Specifies the number of minutes the scheduler on a client node waits between retry attempts after a failed attempt to contact the server or after a scheduled command fails to process. When setting the retry period, set a time period that permits more than one retry attempt within a typical startup window. You can specify an integer from 1 to 9999.

### Example: Set a fifteen minute time period between retry attempts

Have the client scheduler retry failed attempts to contact the server or to process scheduled commands every fifteen minutes.

```
set retryperiod 15
```

### Related commands

Table 325. Commands related to SET RETRYPERIOD

| Command           | Description  |
|-------------------|--|
| QUERY STATUS      | Displays the settings of server parameters, such as those selected by the SET commands.        |
| SET MAXCMDRETRIES | Specifies the maximum number of retries after a failed attempt to execute a scheduled command. |

## SET SCHEDMODES (Select a central scheduling mode)

Use this command to determine how the clients communicate with the server to begin scheduled work. You must configure each client to select the scheduling mode in which it operates.

Use this command with the SET RETRYPERIOD command to regulate the time and the number of retry attempts to process a failed command.

You can issue the QUERY STATUS command to display the value for the scheduling mode supported. At installation, this value is ANY.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### ANY

Specifies that clients can run in either the client-polling or the server-prompted scheduling mode.

#### POLLing

Specifies that only the client-polling mode can be used. Client nodes poll the server at prescribed time intervals to obtain scheduled work.

#### 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.

### Example: Restrict scheduled operations to clients using client-polling

Clients can run under 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 run scheduled commands. This means that any client scheduler using the server-prompted mode waits until you set the schedule mode to ANY or PROMPTED.

```
set schedmodes polling
```

### Related commands

Table 326. Command related to SET SCHEDMODES

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |



*Table 326. Command related to SET SCHEDMODES (continued)*

| Command         | Description  |
|-----------------|--|
| SET RETRYPERIOD | Specifies the time between retry attempts by the client scheduler. |

## SET SERVERHLADDRESS (Set the high-level address of a server)

Use this command to set the high-level address (IP) of a server. IBM Tivoli Storage Manager uses the address when you issue a DEFINE SERVER command with CROSSDEFINE=YES.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set SERVERHladdress—*ip\_address*—————►◄

### Parameters

#### *ip\_address* (Required)

Specifies a server high-level address as a numeric dotted decimal name or a host name. If a host name is specified, a server that can resolve the name to the dotted decimal form must be available.

### Example: Set the high-level address of a server

Set the high-level address of HQ\_SERVER to 9.230.99.66.

```
set serverhladdress 9.230.99.66
```

### Related commands

*Table 327. Command related to SET SERVERHLADDRESS*

| Command             | Description                                  |
|---------------------|--|
| SET CROSSDEFINE     | Specifies whether to cross define servers.   |
| SET SERVERLLADDRESS | Specifies the low-level address of a server. |
| SET SERVERPASSWORD  | Specifies the server password.               |

## SET SERVERLLADDRESS (Set the low-level address of a server)

Use this command to set the low-level address of a server. Tivoli Storage Manager uses the address when you issue a DEFINE SERVER command with CROSSDEFINE=YES.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set SERVERLladdress—*tcp\_port*—————►◄

### Parameters

#### *tcp\_port* (Required)

Specifies the low-level address of the server. Generally, this address is identical to the TCPPOINT option in the server option file of the server.

### Example: Set the low-level address of a server

Set the low-level address of HQ\_SERVER to 1500.

```
set serverlladdress 1500
```

### Related commands

Table 328. Command related to SET SERVERLLADDRESS

| Command             | Description                                   |
|---------------------|---|
| SET CROSSDEFINE     | Specifies whether to cross define servers.    |
| SET SERVERHLADDRESS | Specifies the high-level address of a server. |
| SET SERVERPASSWORD  | Specifies the server password.                |

### SET SERVERNAME (Specify the server name)

Use this command to change the server name. When you install the Tivoli Storage Manager server, the name is set at installation to SERVER1.

Use the QUERY STATUS command to display the server name.

If you migrate from ADSM to Tivoli Storage Manager, the name is set to ADSM or the name last specified to ADSM with a SET SERVERNAME command.

#### Important:

- If this is a source server for a virtual volume operation, changing its name can impact its ability to access and manage the data it has stored on the corresponding target server.
- To prevent problems related to volume ownership, do not change the name of a server if it is a library client.

When changing the name of a server, be aware of the following additional restrictions:

- Windows clients use the server name to identify which passwords belong to which servers. Changing the server name after the clients are connected forces the clients to reenter the passwords.
- You must set unique names on servers that communicate with each other. For details, see the *Administrator's Guide*. On a network where clients connect to multiple servers, it is recommended that all of the servers have unique names.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set SERVERname—*server\_name*—————►►

### Parameters

#### *server\_name* (Required)

Specifies the new server name. The name must be unique across a server network for enterprise event logging, enterprise configuration, command routing, or virtual volumes. The maximum length of the name is 64 characters.

### Example: Name the server

Name the server WELLS\_DESIGN\_DEPT.

```
set servername wells_design_dept
```

### Related commands

Table 329. Command related to SET SERVERNAME

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET SERVERPASSWORD (Set password for server)

Use this command to set the password for communication between servers to support enterprise administration and enterprise event logging and monitoring.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set SERVERPAssword—*password*—————◄◄

### Parameters

#### *password* (Required)

Specifies a password for the server. Other servers must have the same password in their definitions of this server.

### Example: Set a server's password

Set the password for HQ\_SERVER to agave.

```
set serverpassword agave
```

### Related commands

Table 330. Command related to SET SERVERPASSWORD

| Command             | Description                                   |
|---------------------|---|
| SET CROSSDEFINE     | Specifies whether to cross define servers.    |
| SET SERVERHLADDRESS | Specifies the high-level address of a server. |
| SET SERVERLLADDRESS | Specifies the low-level address of a server.  |

## SET SUBFILE (Set subfile backup for client nodes)

Use this command to set up the server to allow clients to back up subfiles. On the client's machine, the SUBFILEBACKUP, SUBFILECACHEPATH, and SUBFILECACHESIZE options must be specified in the client's options file (dsm.opt).

With subfile backups, when a client's file has been previously backed up, any subsequent backups are typically made to the portion (a subfile) of the client's file that has changed, rather than the entire file.

Use the QUERY STATUS command to determine whether subfiles can be backed up to the server running this command.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```

>> Set SUBFILE {Client | No}

```

### Parameters

#### Client

Specifies that the client node can determine whether to use subfile backup.

#### No

Specifies that the subfile backups are not to be used. At installation, this value is set to No.

### Example: Set subfile backup for client nodes

Allow the client node to backup subfiles on the server.

```
set subfile client
```

### Related commands

Table 331. Command related to SET SUBFILE

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SET SUMMARYRETENTION (Set number of days to keep data in activity summary table)

Use this command to specify the number of days to keep information in the SQL activity summary table.

The SQL activity summary table contains statistics about each client session and server processes. For a description of the information in the SQL activity summary table, issue the following command:

```
select colname, remarks from columns where tabname='SUMMARY'
```

Issue the QUERY STATUS command to display the number of days the information is kept. At installation, Tivoli Storage Manager allows each server to determine its own number of days for keeping information in the SQL activity summary table.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set SUMmaryretention—┐
                           └days┘
```

### Parameters

*days*

Specifies the number of days to keep information in the activity summary table. You can specify a number from 0 to 9999. A value of 0 means that information in the activity summary table is not kept. A value of 1 specifies to keep the activity summary table for the current day.

### Example: Specify the number of days to keep information in the SQL activity summary table

Set the server to retain the activity summary table information for 15 days.

```
set summaryretention 15
```

### Related commands

Table 332. Commands related to SET SUMMARYRETENTION

| Command             | Description   |
|---------------------|---|
| QUERY STATUS        | Displays the settings of server parameters, such as those selected by the SET commands. |
| SET ACTLOGRETENTION | Specifies the number of days to retain log records in the activity log.                 |
| QUERY ACTLOG        | Displays messages from the server activity log.   |
| SELECT              | Allows customized queries of the IBM Tivoli Storage Manager database.                   |

## SET TAPEALERTMSG (Set tape alert messages on or off)

Use this command to allow the Tivoli Storage Manager server to log notification of diagnostic information from library and drive devices. At installation, this value is set to OFF. When enabled, Tivoli Storage Manager can retrieve diagnostic information from a tape or library device and display it using ANR messages. When disabled, Tivoli Storage Manager will not query a device for this information.

### Privilege class

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

### Syntax

```
►►—Set TAPEAlertmsg—┐ON┐—————►►
                     └─OFF─┘
```

### Parameters

#### ON

Specifies that diagnostic information will be reported to the Tivoli Storage Manager server.

#### OFF

Specifies that diagnostic information will not be reported to the Tivoli Storage Manager server.

### Example: Set tape alert messages on

Allow the Tivoli Storage Manager server to receive diagnostic information messages.

```
set tapealertmsg on
```

### Related commands

Table 333. Command related to SET TAPEALERTMSG

| Command            | Description   |
|--------------------|---|
| QUERY TAPEALERTMSG | Displays whether the server logs hardware diagnostic information. |



**SET TOCLOADRETENTION (Set load retention period for table of contents)**

Use this command to specify the approximate number of minutes that unreferenced table of contents data will remain loaded in the server database.

During NDMP-controlled backup operations of NAS file systems, the server can optionally collect information about files and directories in the image and store this information in a table of contents within a storage pool. The Web client can be used to examine files and directories in one or more file-system images by displaying entries from the table of contents data. The server loads the necessary table of contents data into a temporary database table.

Once the data have been loaded, the user can then select those files and directories to be restored. Because this database table is temporary, the data will only remain loaded for a specified time since the last reference to that data. At installation, the retention time is set to 120 minutes. Use the QUERY STATUS command to see the table of contents load retention time.

**Privilege class**

To issue this command you must have system privilege.

**Syntax**

►►—Set TOCLOADRetention—*minutes*—————►◄

**Parameters**

*minutes* **(Required)**  
Specifies the approximate number of minutes that an unreferenced table of contents data is retained in the database. You can specify an integer from 30 to 1000.

**Example: Define the load retention period for the table of contents**

Use the command, SET TOCLOADRETENTION, to specify that unreferenced table of contents data is to be retained in the database for 45 minutes.

```
set toclloadretention 45
```

**Related commands**

Table 334. Commands related to SET TOCLOADRETENTION

| Command      | Description   |
|--------------|---|
| QUERY STATUS | Displays the settings of server parameters, such as those selected by the SET commands. |

## SETOPT (Set a server option for dynamic update)

Use this command to update a server option dynamically without stopping and restarting the server. A SETOPT command contained in a macro or a script cannot be rolled back.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—SETOPT—*option\_name*—*option\_value*—►►

### Parameters

#### *option\_name* (Required)

Specifies a text string of information identifying the server option to be updated. The maximum length of the text string is 255 characters. The following options are available:

COMMTIMEOUT  
DEDUPREQUIRESBACKUP  
DNSLOOKUP  
EXPINTERVAL  
EXPQUIET  
IDLETIMEOUT  
MAXSESSIONS  
MOVEBATCHSIZE  
MOVESIZETHRESH  
NDMPREFDATAINTERFACE  
RECLAIMDELAY  
RECLAIMPERIOD  
RESTOREINTERVAL  
RETENTIONEXTENSION  
SANDISCOVERY  
SANREFRESHTIME  
SHREDDING  
THROUGHPUTDATATHRESHOLD  
THROUGHPUTTIMETHRESHOLD  
TXNGROUPMAX

#### *option\_value* (Required)

Specifies the value for the server option.

### Example: Set the maximum number of client session

Update the server option for the maximum number of client sessions to a value of 40.

```
setopt maxsessions 40
```

### Related commands

Table 335. Commands related to SETOPT

| Command      | Description                                |
|--------------|--|
| QUERY OPTION | Displays information about server options. |

*Table 335. Commands related to SETOPT (continued)*

| Command      | Description  |
|--------------|--|
| QUERY SYSTEM | Displays details about the IBM Tivoli Storage Manager server system. |

## SHRED DATA (Shred data)

Use this command to manually start the process of shredding deleted sensitive data. Manual shredding is possible only if automatic shredding is disabled.

You can control automatic shred processing with the SHREDDING server option.

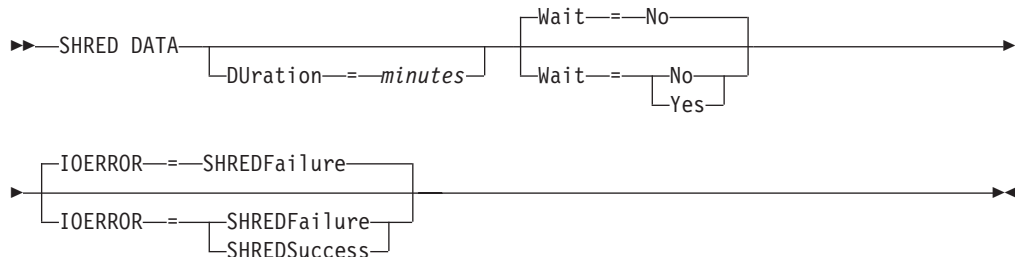
This command creates a background process that can be cancelled with the CANCEL PROCESS command. To display information on background processes, use the QUERY PROCESS command.

If data from a storage pool that enforces shredding is deleted while a manual shredding process is running, it will be added to the running process.

### Privilege class

To issue this command you must have system privilege.

### Syntax



### Parameters

#### DURATION

Specifies the maximum number of minutes the shredding process runs before being automatically cancelled. When the specified number of minutes elapses, the server cancels the shredding process. As soon as the process recognizes the cancellation, it ends. Because of this, the process may run longer than the value you specified for this parameter. You can specify a number from 1 to 9999. This parameter is optional. If not specified, the server will stop only after all deleted sensitive data has been shredded.

#### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is No. Possible values are:

##### No

Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed. Messages created from the background process are displayed either in the activity log or the server console, or both, depending on where messages are logged. To cancel a background process, use the CANCEL PROCESS command. If you cancel this process, some files might already have been shredded before the cancellation. This is the default.

##### Yes

Specifies that the server processes this command in the foreground. You

must wait for the operation to complete before continuing with other tasks. The server displays the output messages to the administrative client when the operation completes. Messages are also displayed either in the activity log or the server console, or both, depending on where messages are logged.

**Note:** You cannot specify WAIT=YES from the server console.

### IOERROR

Specifies whether an I/O error encountered while shredding the data is to be considered a successful shred. This parameter is optional. The default is SHREDFailure. Possible values are:

#### SHREDFailure

Specifies that if the server encounters an I/O error while shredding, the data will not be considered successfully shredded and the owning file will be marked as damaged. The server will attempt to shred the data again the next time the shredding process runs, giving you a chance to correct the error and ensure the data can be properly shredded.

#### SHREDSuccess

Specifies that if the server encounters an I/O error while shredding and the owning file had been previously marked as damaged, the data will be considered successfully shredded. You should use this option only after the server has reported I/O errors while shredding and you are unable to correct the error.

### Example: Shred data

Manually start the shredding of all deleted sensitive data. Continue the process for up to six hours before automatically cancelling it.

```
shred data duration=360
```

### Related commands

*Table 336. Commands related to SHRED DATA*

| Command           | Description   |
|-------------------|---|
| CANCEL PROCESS    | Cancels a background server process.                    |
| QUERY PROCESS     | Displays information about background processes.        |
| QUERY SHREDSTATUS | Displays information about data waiting to be shredded. |

## SUSPEND EXPORT (Suspend a currently running export operation)

Use this command to suspend a currently running server-to-server export operation which has a FILEDATA value that is not NONE. The export operation that you want to suspend must be past the initialization phase to be eligible for suspension. The state of the export operation is saved. The operation can be restarted by issuing the RESTART EXPORT command.

### Privilege class

You must have system privilege to issue this command.

### Syntax



### Parameters

#### EXPORTIDENTIFIER

This optional parameter specifies the name of the export operation. You can find a name by issuing the QUERY EXPORT command to list all the currently running server-to-server export operations that can be suspended. You can also use the wildcard character to specify the name.

### Example: Suspend a specific export operation

Suspend the running export operation EXPORTALLACCTNODES. No output is generated when you issue the SUSPEND EXPORT command. You must issue the QUERY EXPORT command to verify that the EXPORTALLACCTNODES operation is suspended.

```
suspend export exportallacctnodes
```

### Example: Suspend all running export operations

Suspend all the export operations with a state of RUNNING.

```
suspend export *
```

### Related commands

Table 337. Commands related to SUSPEND EXPORT

| Command        | Description   |
|----------------|---|
| CANCEL EXPORT  | Deletes a suspended export operation                                    |
| EXPORT NODE    | Copies client node information to external media.                       |
| EXPORT SERVER  | Copies all or part of the server to external media.                     |
| QUERY EXPORT   | Displays the export operations that are currently running or suspended. |
| RESTART EXPORT | Restarts a suspended export operation.                                  |

---

## UNLOCK commands

Use the UNLOCK command to reaccess a server after having being locked out.

The following is a list of UNLOCK commands for Tivoli Storage Manager:

- “UNLOCK ADMIN (Unlock an administrator)” on page 962
- “UNLOCK NODE (Unlock a client node)” on page 963
- “UNLOCK PROFILE (Unlock a profile)” on page 964

## UNLOCK ADMIN (Unlock an administrator)

Use this command to allow a locked administrator to access the server again.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—UNLOCK Admin—*admin\_name*—◄◄

### Parameters

*admin\_name* (Required)

Specifies the name of the administrator to unlock.

### Example: Unlock an administrator

The administrator JOE is locked out of IBM Tivoli Storage Manager. Permit JOE to access the server. Issue the following command:

```
unlock admin joe
```

### Related commands

Table 338. Commands related to UNLOCK ADMIN

| Command     | Description   |
|-------------|---|
| LOCK ADMIN  | Prevents an administrator from accessing IBM Tivoli Storage Manager.              |
| QUERY ADMIN | Displays information about one or more IBM Tivoli Storage Manager administrators. |



## UNLOCK NODE (Unlock a client node)

Use this command to allow a locked client node to access the server again.

### 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—*node\_name*—◄◄

### Parameters

*node\_name* **(Required)**  
Specifies the name of the client node to unlock.

### Example: Unlock a node

The client node SMITH is locked out of IBM Tivoli Storage Manager. Permit SMITH to access the server.

```
unlock node smith
```

### Related commands

Table 339. Commands related to UNLOCK NODE

| Command    | Description   |
|------------|---|
| LOCK NODE  | Prevents a client from accessing the server.                        |
| QUERY NODE | Displays partial or complete information about one or more clients. |

## UNLOCK PROFILE (Unlock a profile)

Use this command on a configuration manager to unlock a configuration profile so it can be distributed to subscribing managed servers.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—UNLOCK PROFILE—*profile\_name*—◄◄

### Parameters

#### *profile\_name* (Required)

Specifies the profile to unlock. You can use wildcard characters to indicate multiple names.

### Example: Unlock a profile

Unlock a profile named TOM.

```
unlock profile tom
```

### Related commands

Table 340. Commands related to UNLOCK PROFILE

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                       |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                 |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                    |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                  |
| QUERY PROFILE          | Displays information about configuration profiles.                 |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |
| UPDATE PROFILE         | Changes the description of a profile.                              |

---

## UPDATE commands

Use the UPDATE command to modify one or more attributes of an existing Tivoli Storage Manager object.

The following is a list of UPDATE commands for Tivoli Storage Manager:

- “UPDATE ADMIN (Update an administrator)” on page 966
- “UPDATE BACKUPSET (Update a retention value assigned to a backup set)” on page 968
- “UPDATE CLIENTOPT (Update a client option sequence number)” on page 973
- “UPDATE CLOPTSET (Update a client option set description)” on page 974
- “UPDATE COLLOGROUP (Update a collocation group)” on page 975
- “UPDATE COPYGROUP (Update a copy group)” on page 976
- “UPDATE DATAMOVER (Update a data mover)” on page 984
- “UPDATE DEVCLASS (Update the attributes of a device class)” on page 986
- “UPDATE DOMAIN (Update a policy domain)” on page 1053
- “UPDATE DRIVE (Update a drive)” on page 1055
- “UPDATE LIBRARY (Update a library)” on page 1059
- “UPDATE LIBVOLUME (Change the status of a storage volume)” on page 1064
- “UPDATE MACHINE (Update machine information)” on page 1066
- “UPDATE MGMTCLASS (Update a management class)” on page 1068
- “UPDATE NODE (Update node attributes)” on page 1071
- “UPDATE NODEGROUP (Update a node group)” on page 1079
- “UPDATE PATH (Change a path)” on page 1080
- “UPDATE POLICYSET (Update a policy set description)” on page 1085
- “UPDATE PROFILE (Update a profile description)” on page 1087
- “UPDATE RECOVERYMEDIA (Update recovery media)” on page 1088
- “UPDATE SCHEDULE (Update a schedule)” on page 1090
- “UPDATE SCRIPT (Update a Tivoli Storage Manager script)” on page 1110
- “UPDATE SERVER (Update a server defined for server-to-server communications)” on page 1112
- “UPDATE SERVERGROUP (Update a server group description)” on page 1116
- “UPDATE SPACETRIGGER (Update the space triggers)” on page 1117
- “UPDATE STGPOOL (Update a storage pool)” on page 1119
- “UPDATE VIRTUALFSMAPPING (Update a virtual file space mapping)” on page 1153
- “UPDATE VOLHISTORY (Update sequential volume history information)” on page 1155
- “UPDATE VOLUME (Change a storage pool volume)” on page 1157

## UPDATE ADMIN (Update an administrator)

Use this command to change the password or contact information for an administrator. However, you cannot update the SERVER\_CONSOLE administrator name.

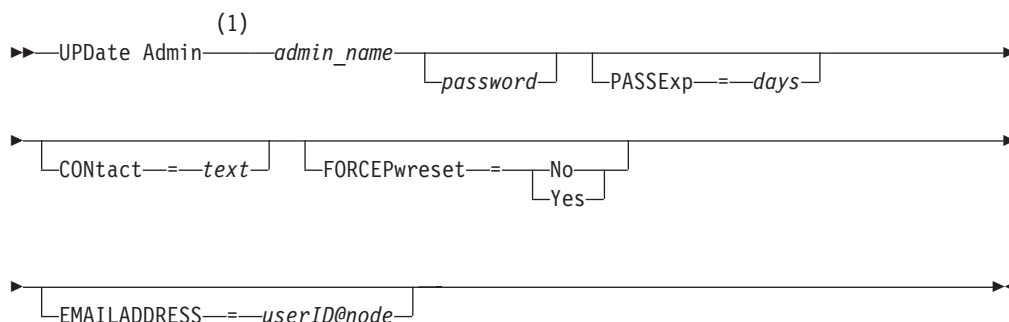
Passwords for administrators must be changed after a length of time determined by the SET PASSEXP command.

You must use the RENAME ADMIN command to change the name of a registered administrator.

### 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



### Notes:

- 1 You must specify at least one optional parameter on this command.

### Parameters

#### *admin\_name* (Required)

Specifies the name of the administrator to be updated.

#### *password*

Specifies the administrator's password. This parameter is optional. Passwords remain current for a period determined by the password expiration period.

#### **PASSExp**

Specifies the number of days the password remains valid. You can set the password expiration period from 0 to 9999 days. A value of 0 means that the password never expires. This parameter is optional. If you do not specify this parameter the password expiration period is unchanged.

#### **CONtact**

Specifies a text string that identifies the administrator. This parameter is optional. Enclose the text string in quotation marks if it contains any blanks. To remove previously defined contact information, specify a null string ("").

#### **FORCEPwreset**

Specifies whether the administrator is required to change or reset the password. This parameter is optional. Possible values are:

**No**

Specifies that the administrator does not need to change or reset the password while attempting to sign on to the server. The password expiration period is set by the SET PASSEXP command.

**Yes**

Specifies that the administrator's password will expire at the next sign on. The administrator must change or reset the password at that time. If a password is not specified, you will receive a syntax error.

**EMAILADDRESS**

This parameter is used by IBM Tivoli Storage Manager Express only. It is not currently used by Tivoli Storage Manager. Tivoli Storage Manager Express uses this parameter to specify the e-mail address or addresses for this administrator. Any reports that are directed to this administrator are sent to these e-mail addresses.

**Example: Update a password and password expiration period**

Update the administrator LARRY to have the password SECRETWORD and a password expiration period of 120 days.

```
update admin larry secretword passexp=120
```

**Related commands**

*Table 341. Commands related to UPDATE ADMIN*

| Command        | Description   |
|----------------|---|
| QUERY ADMIN    | Displays information about one or more IBM Tivoli Storage Manager administrators.       |
| QUERY STATUS   | Displays the settings of server parameters, such as those selected by the SET commands. |
| REGISTER ADMIN | Defines a new administrator without granting administrative authority.                  |
| REGISTER NODE  | Defines a client to the server and sets options for that user.                          |
| RENAME ADMIN   | Changes an IBM Tivoli Storage Manager administrator's name.                             |
| SET PASSEXP    | Specifies the number of days after which a password is expired and must be changed.     |
| UPDATE NODE    | Changes the attributes associated with a client node.                                   |

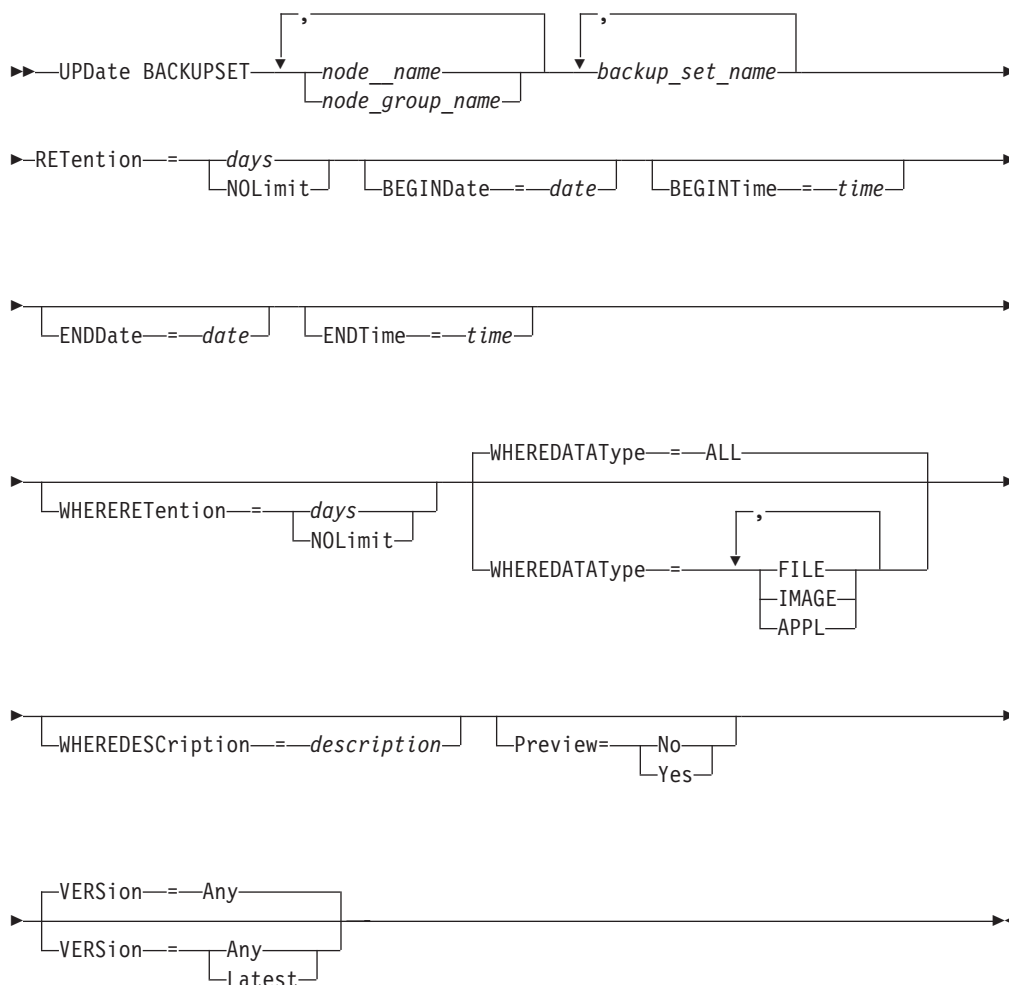
## UPDATE BACKUPSET (Update a retention value assigned to a backup set)

Use this command to update the retention value associated with a client's backup set.

### Privilege class

To issue this command, you must have system privilege or policy privilege for the domain to which the client node is assigned.

### Syntax



### Parameters

#### *node\_name* or *node\_group\_name* (Required)

Specifies the names of the client nodes or node groups whose data is contained in the specified backup set to be updated. To specify multiple node and node group names, separate the names with commas and no intervening spaces. The node names that you specify can contain wildcard characters, but node group names cannot contain wildcard characters.

***backup\_set\_name* (Required)**

Specifies the name of the backup set to update. The backup set name you specify can contain wildcard characters. You can specify more than one backup set name by separating the names with commas and no intervening spaces.

**RETention (Required)**

Specifies the updated number of days to retain the backup set on the server. You can specify an integer from 0 to 30000. The values are:

*days*

Specifies the updated number of days to retain the backup set.

**NOLimit**

Specifies that the backup set is retained on the server indefinitely. If you specify NOLIMIT, the server retains the volumes containing the backup set forever, unless a user or administrator deletes the volumes from server storage.

**Attention:** Updating the retention period of a backup set may cause it to expire at a different time from other backup sets that might be stored on the same output media. In either case, the media will not be made available for other uses until all of its backup sets have expired.

**BEGINDate**

Specifies the beginning date in which the backup set to update was created. This parameter is optional. The default is the current date. You can use this parameter with the **BEGINTIME** parameter to specify a range for the date and time. If you specify a begin date without a begin time, the time will be at 12:00 a.m. (midnight) on the date you specify.

You can specify the date by using one of the following values:

| Value                                  | Description                            | Example         |
|--|--|-----------------|
| <i>MM/DD/YYYY</i>                      | A specific date                        | 09/15/1999      |
| TODAY                                  | The current date                       | TODAY           |
| TODAY+ <i>days</i> or<br>+ <i>days</i> | The current date plus days specified.  | TODAY +3 or +3. |
| TODAY- <i>days</i> or<br>- <i>days</i> | The current date minus days specified. | TODAY-3 or -3.  |

**BEGINTime**

Specifies the beginning time in which the backup set to update was created. This parameter is optional. The default is the current time. You can use this parameter with the **BEGINDATE** parameter to specify a range for the date and time. If you specify a begin time without a begin date, the date will be the current date at the time you specify.

You can specify the time by using one of the following values:

| Value                                  | Description   | Example              |
|--|---|----------------------|
| <i>HH:MM:SS</i>                        | A specific time   | 10:30:08             |
| NOW                                    | The current time  | NOW                  |
| NOW+ <i>HH:MM</i><br>or + <i>HH:MM</i> | The current time plus hours and minutes on the specified end date | NOW+02:00 or +02:00. |

## UPDATE BACKUPSET

| Value                                  | Description  | Example              |
|--|--|----------------------|
| NOW- <i>HH:MM</i><br>or - <i>HH:MM</i> | The current time minus hours and minutes on the specified end date | NOW-02:00 or -02:00. |

### ENDDate

Specifies the ending date in which the backup set to update was created. This parameter is optional. You can use this parameter with the **ENDTIME** parameter to specify a range for the date and time. If you specify an end date without an ending time, the time will be at 11:59:59 p.m. on the specified end date.

You can specify the date by using one of the following values:

| Value                                  | Description                            | Example         |
|--|--|-----------------|
| <i>MM/DD/YYYY</i>                      | A specific date                        | 09/15/1999      |
| TODAY                                  | The current date                       | TODAY           |
| TODAY+ <i>days</i> or<br>+ <i>days</i> | The current date plus days specified.  | TODAY +3 or +3. |
| TODAY- <i>days</i> or<br>— <i>days</i> | The current date minus days specified. | TODAY -3 or -3. |

### ENDTime

Specifies the ending time in which the backup set to update was created. This parameter is optional. You can use this parameter with the **ENDDATE** parameter to specify a range for the date and time. If you specify an end time without an end date, the date will be the current date at the time you specify.

You can specify the time by using one of the following values:

| Value                                  | Description  | Example              |
|--|--|----------------------|
| <i>HH:MM:SS</i>                        | A specific time                                    | 10:30:08             |
| NOW                                    | The current time                                   | NOW                  |
| NOW+ <i>HH:MM</i><br>or + <i>HH:MM</i> | The current time plus hours and minutes specified  | NOW+02:00 or +02:00. |
| NOW- <i>HH:MM</i><br>or - <i>HH:MM</i> | The current time minus hours and minutes specified | NOW-02:00 or -02:00. |

### WHERERETention

Specifies the retention value, specified in days, that is associated with the backup set to update. The values are:

*days*

Specifies that the backup set that is retained this number of days is updated.

### NOLimit

Specifies that the backup set retained indefinitely is updated.

### WHEREDESCription

Specifies the description that is associated with the backup set to update. This parameter is optional. You can specify wildcard characters for the description. Enclose the description in quotation marks if it contains any blank characters.

### WHEREDATAType

Specifies the backup sets containing the specified types of data are to be



updated. This parameter is optional. The default is that backup sets for all types of data (file level, image, and application) are to be updated. To specify multiple data types, separate each data type with a comma and no intervening spaces. Possible values are:

**ALL**

Specifies that backup sets for all types of data (file level, image, and application) are to be updated. This is the default.

**FILE**

Specifies that a file level backup set is to be updated. File level backup sets contain files and directories backup up by the backup-archive client.

**IMAGE**

Specifies that an image backup set is to be updated. Image backup sets contain images created by the backup-archive client BACKUP IMAGE command.

**APPL**

Specifies that an application backup set is to be updated. Application backup sets contain data backed up to the Tivoli Storage Manager server by applications using the Tivoli Storage Manager application programming interface. This parameter applies only if the Tivoli Storage Manager server was upgraded from Tivoli Storage Manager Express and only for the Tivoli Storage Manager Express versions of Data Protection for Exchange and Data Protection for SQL.

**Preview**

Specifies whether to preview the list of backup sets to update, without actually updating the backup sets. This parameter is optional. The default is No. The values are:

**No**

Specifies that the backup sets are updated.

**Yes**

Specifies that the server displays the backup sets to update, without actually updating the backup sets.

**VERSION**

Specifies the version of the backup set to update. Backup sets with the same prefix name are considered to be different versions of the same backup set. This parameter is optional. The default is to update any version that matches the criteria specified on the command. The values are:

**Any**

Specifies that any version that matches the criteria specified on the command should be updated.

**Latest**

Specifies that only the most recent version of the backup set should be updated. If other criteria specified on the command (for example, ENDDATE or WHERERETENTION) exclude the most recent version of the backup set, then no backup set will be updated.

**Example: Update a retention period**

Update the retention period where the description is Healthy Computers. The retention period is assigned to backup set PERS\_DATA.3099 that contains data from client node JANE. Change the retention period to 70 days.

## UPDATE BACKUPSET

```
update backupset jane pers_data.3099  
retention=70 wheredescription="healthy computers"
```

### Related commands

*Table 342. Commands related to UPDATE BACKUPSET*

| Command                 | Description   |
|-------------------------|---|
| DEFINE BACKUPSET        | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP        | Defines a group of nodes.                               |
| DEFINE NODEGROUPMEMBER  | Adds a client node to a node group.                     |
| DELETE BACKUPSET        | Updates a retention value associated with a backup set. |
| DELETE NODEGROUP        | Deletes a node group.                                   |
| DELETE NODEGROUPMEMBER  | Deletes a client node from a node group.                |
| GENERATE BACKUPSET      | Generates a backup set of a client's data.              |
| GENERATE BACKUPSETTOC   | Generates a table of contents for a backup set.         |
| QUERY BACKUPSET         | Displays backup sets.                                   |
| QUERY BACKUPSETCONTENTS | Displays contents contained in backup sets.             |
| QUERY NODEGROUP         | Displays information about node groups.                 |
| UPDATE NODEGROUP        | Updates the description of a node group.                |

## UPDATE CLIENTOPT (Update a client option sequence number)

Use this command to update the sequence number of a client option in a client option set.

### Privilege class

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

### Syntax

```
►►—UPDate CLIENTOpt—option_set_name—option_name—————►
►—current_sequence_number—new_sequence_number—————►◄
```

### Parameters

*option\_set\_name* **(Required)**

Specifies the name of the option set.

*option\_name* **(Required)**

Specifies a valid client option.

*current\_sequence\_number* **(Required)**

Specifies the current sequence number of the option.

*new\_sequence\_number* **(Required)**

Specifies the new sequence number of the option.

### Example: Update a client option sequence number

To update the current client option sequence number issue the following command:

```
update clientopt eng dateformat 0 9
```

### Related commands

*Table 343. Commands related to UPDATE CLIENTOPT*

| Command          | Description                                       |
|------------------|---|
| COPY CLOPTSET    | Copies a client option set.                       |
| DEFINE CLIENTOPT | Adds a client option to a client option set.      |
| DELETE CLIENTOPT | Deletes a client option from a client option set. |
| DELETE CLOPTSET  | Deletes a client option set.                      |
| QUERY CLOPTSET   | Displays information about a client option set.   |

## UPDATE CLOPTSET (Update a client option set description)

Use this command to update the description for a client option 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 client node is assigned.

### Syntax

```
►►—UPDate CLOptset—option_set_name—DESCription—==description—►◄
```

### Parameters

#### *option\_set\_name* (Required)

Specifies the name of the option set.

#### DESCription (Required)

Specifies a description of the client option set. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains blank characters.

### Example: Update a client option set description

Update the description for a client option set named ENG.

```
update cloptset eng description="unix"
```

### Related commands

Table 344. Commands related to UPDATE CLOPTSET

| Command          | Description  |
|------------------|--|
| COPY CLOPTSET    | Copies a client option set.  |
| DEFINE CLIENTOPT | Adds a client option to a client option set.                           |
| DEFINE CLOPTSET  | Defines a client option set.   |
| DELETE CLIENTOPT | Deletes a client option from a client option set.                      |
| DELETE CLOPTSET  | Deletes a client option set.   |
| QUERY CLOPTSET   | Displays information about a client option set.                        |
| UPDATE CLIENTOPT | Updates the sequence number of a client option in a client option set. |

## UPDATE COLLOCGROUP (Update a collocation group)

Use this command to modify the description of a collocation group.

### Privilege class

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

### Syntax

```
►►—UPDate COLLOCGroup—group_name—DESCription—=—description—◄◄
```

### Parameters

*group\_name*

Specifies the name of the collocation group whose description you want to update.

**DESCription**

Specifies a description of the collocation group. This parameter is required. The maximum length of the description is 255 characters. If the description contains any blanks, enclose the entire description in quotation marks.

### Example: Update a collocation group

Update the collocation group, GROUP1, with a new description.

```
update collogroup group1 "Human Resources"
```

### Related commands

Table 345. Commands related to UPDATE COLLOCGROUP

| Command             | Description  |
|---------------------|--|
| DEFINE COLLOCGROUP  | Defines a collocation group.   |
| DEFINE COLLOCMEMBER | Adds a client node to a collocation group.                                       |
| DEFINE STGPOOL      | Defines a storage pool as a named collection of server storage media.            |
| DELETE COLLOCGROUP  | Deletes a collocation group.   |
| DELETE COLLOCMEMBER | Deletes a client node from a collocation group.                                  |
| MOVE NODEDATA       | Moves data for one or more nodes, or a single node with selected file spaces.    |
| QUERY COLLOCGROUP   | Displays information about collocation groups.                                   |
| QUERY NODE          | Displays partial or complete information about one or more clients.              |
| QUERY NODEDATA      | Displays information about the location and size of data for a client node.      |
| QUERY STGPOOL       | Displays information about storage pools.  |
| REMOVE NODE         | Removes a client from the list of registered nodes for a specific policy domain. |
| UPDATE STGPOOL      | Changes the attributes of a storage pool.  |

## UPDATE COPYGROUP (Update a copy group)

Use this command to update a backup or archive copy group. To allow clients to use the updated copy group, you must activate the policy set that contains the copy group.

**Tip:** The UPDATE COPYGROUP command fails if you specify a copy storage pool 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 and parameters for each form are defined separately.

*Table 346. Commands related to UPDATE COPYGROUP*

| Command             | Description  |
|---------------------|--|
| ACTIVATE POLICYSET  | Validates and activates a policy set.  |
| ASSIGN DEFMGMTCLASS | Assigns a management class as the default for a specified policy set.                      |
| COPY MGMTCLASS      | Creates a copy of a management class.  |
| DEFINE COPYGROUP    | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE MGMTCLASS    | Defines a management class.  |
| DELETE COPYGROUP    | Deletes a backup or archive copy group from a policy domain and policy set.                |
| DELETE MGMTCLASS    | Deletes a management class and its copy groups from a policy domain and policy set.        |
| EXPIRE INVENTORY    | Manually starts inventory expiration processing.   |
| QUERY COPYGROUP     | Displays the attributes of a copy group.   |
| QUERY MGMTCLASS     | Displays information about management classes.   |

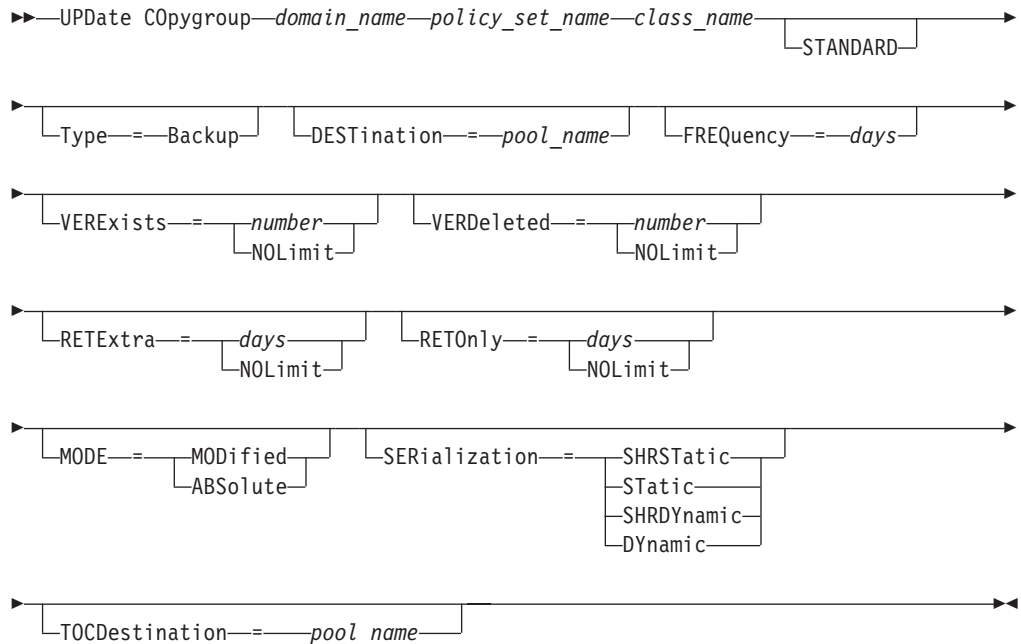
## UPDATE COPYGROUP (Update a backup copy group)

Use this command to update a defined backup 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.

### Syntax



### Parameters

#### *domain\_name* (Required)

Specifies the policy domain to which the copy group belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set to which the copy group belongs. You cannot update a copy group in the ACTIVE policy set.

#### *class\_name* (Required)

Specifies the management class to which the copy group belongs.

#### STANDARD

Specifies the copy group, which must be STANDARD. This parameter is optional.

#### Type=Backup

Specifies that you want to update a backup copy group. This parameter is optional.

#### DESTINATION

Specifies the primary storage pool where the server initially stores backup data. This parameter is optional. You cannot specify a copy storage pool as the destination.

### **FREQuency**

Specifies how frequently the server can back up a file. This parameter is optional. The server backs up a file only when the specified number of days has elapsed since the last backup. The FREQUENCY value is used only during a full incremental backup operation. This value is ignored during selective backup or partial incremental backup. You can specify an integer from 0 to 9999. The value 0 means that the server can back up a file regardless of when the file was last backed up.

### **VERExists**

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

If an incremental backup causes the limit to be exceeded, the server expires the oldest backup version that exists in server storage. Possible values are:

*number*

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 to 9999.

### **NOLimit**

Specifies that you want the server to retain all backup versions.

The number of backup versions to retain is controlled by this parameter until versions exceed the retention time specified by the RETEXTRA parameter.

### **VERDeleted**

Specifies the maximum number of backup versions to retain for files that have been deleted from the client file system after being backed up using the server. This parameter is optional.

If a user deletes a file from the client file system, the next incremental backup causes the server to change the active backup version of the file to inactive and expire the oldest versions in excess of this number. The expiration date for the remaining versions is determined by the retention time specified by the RETEXTRA or RETONLY parameter. Possible values are:

*number*

Specifies the number of backup versions to retain for files that are deleted from the client file system after being backed up. You can specify a value from 0 to 9999.

### **NOLimit**

Specifies that you want the server to retain all backup versions for files that are deleted from the client file system after being backed up.

### **RETEExtra**

Specifies the number of days that the server retains a backup version after that version becomes inactive. A version of a file becomes inactive when the client stores a more recent backup version, or when the client deletes the file from the workstation and then runs a full incremental backup. The server deletes inactive versions based on retention time even if the number of inactive versions does not exceed the number allowed by the VEREXISTS or VERDELETED parameters. This parameter is optional. Possible values are:

*days*

Specifies the number of days to retain inactive backup versions. You can specify an integer from 0 to 9999.

### **NOLimit**

Specifies that you want to retain inactive backup versions indefinitely.



If you specify NOLIMIT, the server deletes extra backup versions based on the VEREXISTS parameter (when the file still exists on the client file system) or the VERDELETED parameter (when the file no longer exists on the client file system).

### **RETOOnly**

Specifies the number of days to retain the last backup version of a file that has been deleted from the client file system. This parameter is optional. Possible values are:

*days*

Specifies the number of days to retain the last remaining inactive copy of a file. You can specify an integer from 0 to 9999.

### **NOLimit**

Specifies that you want to keep the last remaining inactive version of a file indefinitely.

If you specify NOLIMIT, the server retains the last remaining backup version forever, unless a user or administrator deletes the file from server storage.

### **MODE**

Specifies whether the server backs up a file only if the file has changed since the last backup, or whenever a client requests a 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.

The MODE value is used only for full incremental backup. This value is ignored during partial incremental backup or selective backup.

### **SERialization**

Specifies how the server processes files or directories when they are modified during backup processing. This parameter is optional. Possible values are:

#### **SHRStatic**

Specifies that the server backs up a file or directory only if it is not being modified during backup. The server attempts to perform a backup as many as four times, depending on the value specified for the CHANGINGRETRIES client option. If the file or directory is modified during each backup attempt, the server does not back it up.

#### **STatic**

Specifies that the server backs up a file or directory only if it is not being modified during backup. The server attempts to perform the backup only once.

Platforms that do not support the STATIC option default to SHRSTATIC.

### SHRDynamic

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

### DYnamic

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

**Important:** Be careful about using the SHRDynamic and DYnamic values. Tivoli Storage Manager uses these values to determine if it backs up a file or directory while modifications are occurring. As a result, the backup version might be a fuzzy backup. A fuzzy backup does not accurately reflect what is currently in the file or directory because it contains some, but not all, modifications. If a file that contains a fuzzy backup is restored, the file may or may not be usable, depending on the application that uses the file. If a fuzzy backup is not acceptable, set SERIALIZATION to SHRSTATIC or STATIC so that Tivoli Storage Manager creates a backup version only if the file or directory is not being modified.

### TOCDestination

Specifies the primary storage pool in which a table of contents (TOC) will initially be stored for any NDMP backup or backup set operation for which a TOC is generated. This parameter is optional. You cannot specify a copy storage pool as the destination. The storage pool specified for the destination must have NATIVE or NONBLOCK data format. To avoid mount delays, it is recommended that the storage pool have a device class of DISK or DEVTYPE=FILE. TOC generation is an option for NDMP backup operations, but is not supported for other image-backup operations.

To remove an existing TOC destination from the copy group, specify a null string ("" ) for this value.

If TOC creation is requested for a backup operation that uses NDMP and the image is bound to a management class whose backup copy group does not specify a TOC destination, the outcome will depend on the TOC parameter for the backup operation.

- If TOC=PREFERRED (the default), the backup proceeds without creation of a TOC.
- If TOC=YES, the entire backup fails because no TOC can be created.

### Example: Update a backup copy group

Update the backup copy group (STANDARD) in the EMPLOYEE\_RECORDS policy domain, VACATION policy set, ACTIVEFILES management class. Change the destination to DISKPOOL, with a minimum interval of seven days between backups, regardless of whether the files have been modified. Retain up to three backup versions while a file still exists on a client file system.

```
update copygroup employee_records vacation
activefiles type=backup destination=diskpool
frequency=7 verexists=3 mode=absolute
```

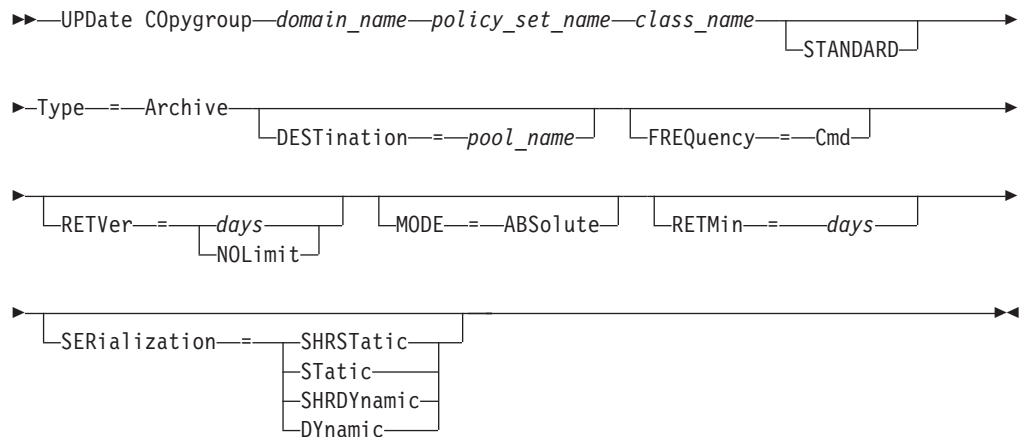
## UPDATE COPYGROUP (Update a defined archive copy group)

Use this command to update a defined 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.

### Syntax



### Parameters

#### *domain\_name* (Required)

Specifies the policy domain to which the copy group belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set to which the copy group belongs. You cannot update a copy group in the ACTIVE policy set.

#### *class\_name* (Required)

Specifies the management class to which the copy group belongs.

#### STANDARD

Specifies the copy group, which must be **STANDARD**. This parameter is optional.

#### Type=Archive

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

#### DESTINATION

Specifies the primary storage pool where the server initially stores the archive copy. This parameter is optional. You cannot specify a copy storage pool as the destination.

#### FREQUENCY=Cmd

Specifies the copy frequency, which must be **CMD**. This parameter is optional.

#### RETVer

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

### *days*

Specifies the number of days to keep an archive copy. You can specify an integer from 0 to 30000.

### **NOLimit**

Specifies that you want to keep an archive copy indefinitely.

If you specify **NOLIMIT**, the server retains archive copies forever, unless a user or administrator deletes the file from server storage.

The value of the **RETV** parameter can affect the management class to which the server binds an archived directory. If the client does not use the ARCHMC option, the server binds directories that are archived to the default management class. If the default management class has no archive copy group, the server binds directories that are archived to the management class with the shortest retention period.

### **MODE=ABSolute**

Specifies that a file is always archived when the client requests it. The MODE must be ABSOLUTE. This parameter is optional.

### **RETM**

Specifies the minimum number of days to keep an archive copy after it has been archived. This parameter is optional. The default value is 365.

### **SERialization**

Specifies how the server processes files that are modified during archive. This parameter is optional. Possible values are:

#### **SHRStatic**

Specifies that the server does not archive a file that is being modified. The server attempts to perform an archive as many as four times, depending on the value specified for the CHANGINGRETRIES client option. If the file is modified during the archive attempt, the server does not archive the file.

#### **Static**

Specifies that the server does not archive a file that is being modified. If a file is modified during the archive attempt, the server does not archive the file.

Platforms that do not support the STATIC option default to SHRSTATIC.

#### **SHRDynamic**

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

#### **Dynamic**

Specifies that the server archives a file on the first attempt, regardless of whether the file is being modified during archive processing.

**Important:** Be careful about using the SHRDYNAMIC and DYNAMIC values. Tivoli Storage Manager uses them to determine if it archives a file while modifications are occurring. As a result, the archive copy might be a fuzzy backup. A fuzzy backup does not accurately reflect what is currently in the file because it contains some, but not all, modifications. If a file that contains a fuzzy backup is retrieved, the file may or may not be usable, depending on the application that uses the file. If a fuzzy backup is not acceptable, set

SERIALIZATION to SHRSTATIC or STATIC so that Tivoli Storage Manager creates an archive copy only if the file is not being modified.

**Tip:** Be cautious when selecting retention values for primary storage pools that are of type RECLAMATIONTYPE=SNAPLOCK. Volumes in these types of storage pools cannot be deleted until after their retention dates have passed.

### Example: Update multiple elements of a copy group

Update the archive copy group (STANDARD) in the EMPLOYEE\_RECORDS policy domain, VACATION policy set, ACTIVEFILES management class. Change the destination to TAPEPOOL. Keep archive copies for 190 days.

```
update copygroup employee_records vacation
activefiles standard type=archive
destination=tapepool retver=190
```

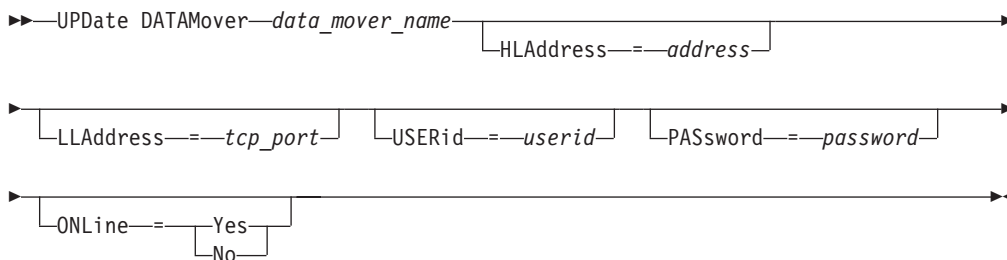
### UPDATE DATAMOVER (Update a data mover)

Use this command to update the definition for a data mover or set a data mover off-line when the hardware is being maintained.

#### Privilege class

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

#### Syntax



#### Parameters

##### *data\_mover\_name* (Required)

Specifies the name of the data mover.

##### HLAddress

Specifies either the new numerical IP address or the new domain name, which is used to access the NAS file server. This parameter is optional.

##### LLAddress

Specifies the new TCP port number to access the NAS file server for Network Data Management Protocol (NDMP) sessions. This parameter is optional.

##### USERid

Specifies the user ID for a user that is authorized to initiate an NDMP session with the NAS file server. For example, enter the administrative ID for a NetApp file server. This parameter is optional.

##### PASsword

Specifies the new password for the user ID to log onto the NAS file server. This parameter is optional.

##### ONLine

Specifies whether the data mover is available for use. This parameter is optional.

##### Yes

Specifies that the data mover is available for use.

##### No

Specifies that the data mover is not available for use.

**Attention:** If a library is controlled using a path from a data mover to the library, and the data mover is offline, the server will not be able to access the library. If the server is halted and restarted while the data mover is offline, the library will not be initialized.

**Example: Update a data mover IP address**

Update the data mover for the node named NAS1. Change the numerical IP address from 9.67.97.103 to 9.67.97.109.

```
update datamover nas1 hladdress=9.67.97.109
```

**Example: Update a data mover domain name**

Update the data mover for the node named NAS1. Change the numerical IP address from 9.67.97.109 to the domain name of NETAPP2.TUCSON.IBM.COM.

```
update datamover nas1 hladdress=netapp2.tucson.ibm.com
```

**Related commands**

*Table 347. Commands related to UPDATE DATAMOVER*

| Command          | Description  |
|------------------|--|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server. |
| DEFINE PATH      | Defines a path from a source to a destination.                 |
| DELETE DATAMOVER | Deletes a data mover.  |
| QUERY DATAMOVER  | Displays data mover definitions.                               |
| REGISTER NODE    | Defines a client to the server and sets options for that user. |
| UPDATE NODE      | Changes the attributes associated with a client node.          |

## UPDATE DEVCLASS (Update the attributes of a device class)

Use this command to update a defined device class.

**Note:** The DISK device class is predefined by IBM Tivoli Storage Manager and cannot be modified with the UPDATE DEVCLASS command.

The syntax and parameter descriptions are provided according to the device type. The syntax and parameter information is presented in the following order:

- 3570 (“Syntax” on page 987)
- 3590 (“Syntax” on page 990)
- 3592 (“Syntax” on page 994)
- 4MM (“Syntax” on page 998)
- 8MM (“Syntax” on page 1002)
- CARTRIDGE (“Syntax” on page 1007)
- CENTERA (“Syntax” on page 1010)
- DLT (“Syntax” on page 1012)
- DTF (“Syntax” on page 1017)
- ECARTRIDGE (“Syntax” on page 1020)
- FILE (“Syntax” on page 1025)
- GENERICTAPE (“Syntax” on page 1029)
- LTO (“Syntax” on page 1031)
- NAS (“Syntax” on page 1036)
- OPTICAL and WORM Types (“Syntax” on page 1038)
- QIC (“Syntax” on page 1042)
- REMOVABLEFILE (“Syntax” on page 1046)
- SERVER (“Syntax” on page 1048)
- VOLSAFE (“Syntax” on page 1050)

*Table 348. Commands related to UPDATE DEVCLASS*

| Command          | Description   |
|------------------|---|
| BACKUP DEVCONFIG | Backs up IBM Tivoli Storage Manager device information to a file. |
| DEFINE DEVCLASS  | Defines a device class.   |
| DEFINE LIBRARY   | Defines an automated or manual library.                           |
| DELETE DEVCLASS  | Deletes a device class name.                                      |
| QUERY DEVCLASS   | Displays information about device classes.                        |
| QUERY DIRSPACE   | Displays information about FILE directories.                      |
| UPDATE LIBRARY   | Changes the attributes of a library.                              |



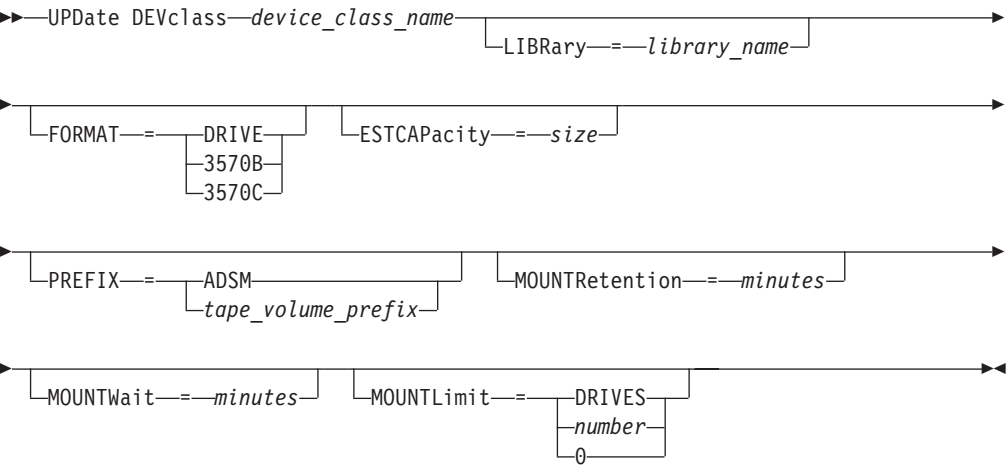
UPDATE DEVCLASS (Update a 3750 device class)

Use the 3570 device class when you are using 3570 tape devices.

Privilege class

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

Syntax



Parameters

device\_class\_name (Required)

Specifies the name of the device class to be defined.

LIBRARY

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for 3570 devices:

Table 349. Recording format and default estimated capacity for 3570 tape volumes

| Format | Estimated Capacity | Description  |
|--------|--------------------|--|
| DRIVE  | —                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3570B  | 5.0 GB             | Uncompressed (basic) format  |

Table 349. Recording format and default estimated capacity for 3570 tape volumes (continued)

| Format | Estimated Capacity | Description       |
|--------|--------------------|-------------------|
| 3570C  | See note10.0 GB    | Compressed format |

**Note:** If 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.

### ESTCAPacity

Specifies the estimated capacity for the 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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

You must specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes).

For more information on the default estimated capacity for 3570 cartridge tapes, see Table 349 on page 987.

### PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### MOUNTRetention

Specifies the number of minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

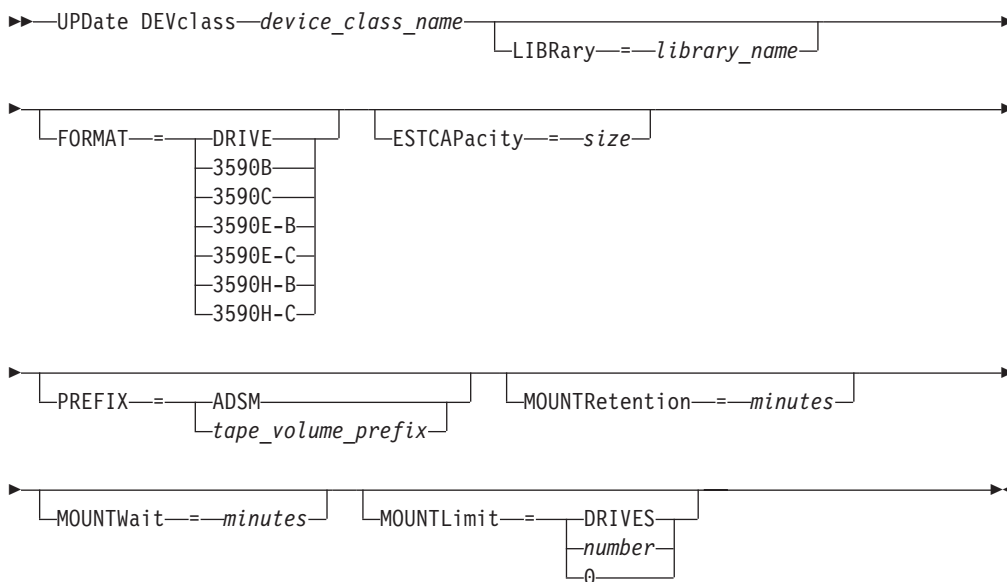
Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

**UPDATE DEVCLASS (Update a 3590 device class)**

Use the 3590 device class when you are using 3590 tape devices.

**Privilege class**

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

**Syntax****Parameters*****device\_class\_name* (Required)**

Specifies the name of the device class to be defined.

**LIBRARY**

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following tables list the recording formats, estimated capacities and recording format options for 3590 devices:

Table 350. Recording formats and default estimated capacities for 3590

| Format  | Estimated Capacity  | Description  |
|---------|---|--|
| DRIVE   | –   | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3590B   | 10.0 GB   | Uncompressed (basic) format  |
| 3590C   | See note<br>20.0 GB   | Compressed format  |
| 3590E-B | 10.0 GB   | Uncompressed (basic) format, similar to the 3590B format   |
| 3590E-C | See note<br>20.0 GB   | Compressed format, similar to the 3590C format   |
| 3590H-B | 30.0 GB (J cartridge-standard length)<br><br>60.0 GB (K cartridge-extended length)                  | Uncompressed (basic) format, similar to the 3590B format   |
| 3590H-C | See note<br><br>60.0 GB (J cartridge-standard length)<br><br>120.0 GB (K cartridge-extended length) | Compressed format, similar to the 3590C format   |

**Note:** If 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.

Table 351. 3590 device recording format selections

| Device     | Format     |            |            |            |            |            |
|------------|------------|------------|------------|------------|------------|------------|
|            | 3590B      | 3590C      | 3590E-B    | 3590E-C    | 3590H-B    | 3590H-C    |
| 3590       | Read/Write | Read/Write | –          | –          | –          | –          |
| Ultra SCSI | Read/Write | Read/Write | –          | –          | –          | –          |
| 3590E      | Read       | Read       | Read/Write | Read/Write | –          | –          |
| 3590H      | Read       | Read       | Read       | Read       | Read/Write | Read/Write |

**ESTCAPacity**

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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

#### **PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

Values specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can contain up to 8 characters (including periods). For example, the following value would be acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetical or national (@,#,\$), followed by alphabetical, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

#### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

#### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

#### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

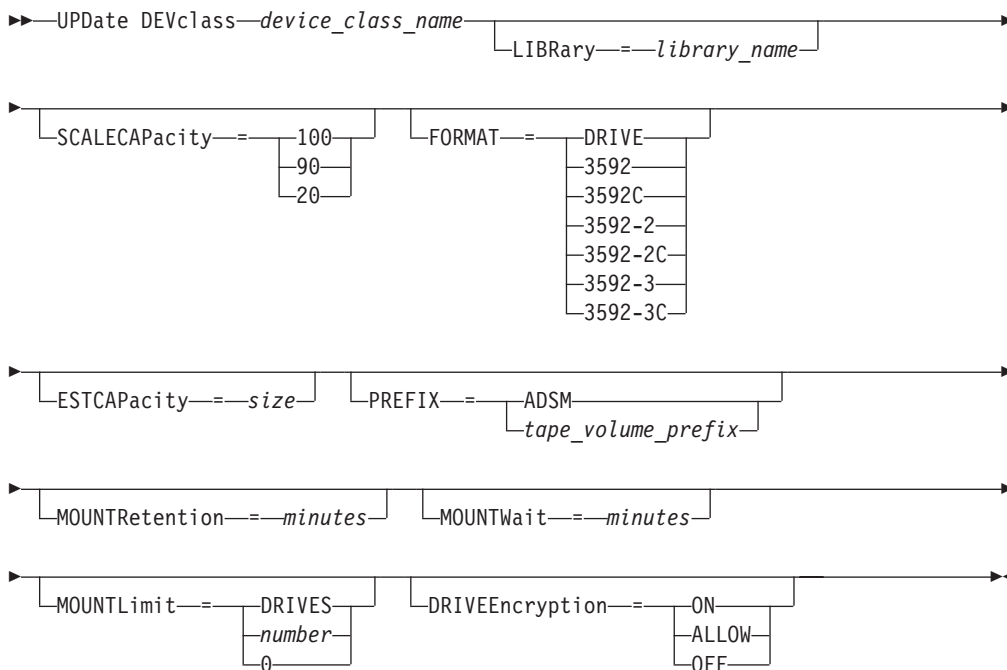
Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete, but new transactions will be terminated.

## UPDATE DEVCLASS (Update a 3592 device class)

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be updated. The maximum length of the device class name is 30 characters.

#### LIBRARY

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

#### SCALECAPacity

Specifies the percentage of the media capacity that can be used to store data. This parameter is optional. Possible values are 20, 90, or 100.

Setting the scale capacity percentage to 100 provides maximum storage capacity. Setting it to 20 provides fastest access time.

**Note:** The scale capacity value will only take effect when data is first written to a volume. Any updates to the device class for scale capacity will not affect volumes that already have data written to them until the volume is returned to scratch status.

#### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.



If the drives are in a library that includes drives of different tape technology, do not use the DRIVE value. Use the specific format that the drives use.

Refer to the *Administrator's Guide* for more information.

The following table lists the recording formats, estimated capacities and recording format options for 3592 devices:

*Table 352. Recording formats and default estimated capacities for 3592*

| Format  | Estimated Capacity | Description  |
|---------|--------------------|--|
| DRIVE   | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3592    | 300 GB             | Uncompressed (basic) format  |
| 3592C   | See note<br>900 GB | Compressed format  |
| 3592-2  | 500 GB<br>700 GB   | Uncompressed (basic) format JA tapes<br>Uncompressed (basic) format JB tapes   |
| 3592-2C | 1.5 TB<br>2.1 TB   | Compressed format JA tapes<br>Compressed format JB tapes   |
| 3592-3  | 640 GB<br>1 TB     | Uncompressed (basic) format JA tapes<br>Uncompressed (basic) format JB tapes   |
| 3592-3C | 1.9 TB<br>3 TB     | Compressed format JA tapes<br>Compressed format JB tapes   |

**Note:** If this format uses the tape-drive hardware-compression feature, depending on the effectiveness of compression, the actual capacity might be different than the listed value.

**Important:** For optimal performance, avoid mixing different generations of drives in a single SCSI library. If you must mix drive generations in an SCSI library, use one of the special configurations that are described in the *Administrator's Guide* to prevent or minimize media problems.

Special configurations are also required for mixing different generations of 3592 drives in 349x and ACSLS libraries. For details, see the *Administrator's Guide*.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

#### **PREFIX**

Specifies the high-level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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, use a volume prefix that conforms to your naming conventions.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

#### **MOUNTRetention**

Specifies the amount of time, in minutes, to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

#### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

#### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device

class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

**DRIVEEncryption**

Specifies whether drive encryption is permitted. This parameter is optional.

Updating this parameter will affect empty volumes only. If a filling volume was previously encrypted or is currently unencrypted, and you update the DRIVEENCRYPTION parameter, the volume maintains its original encrypted or unencrypted status. The filling volume also maintains its original key-management status.

**ON**

Specifies that Tivoli Storage Manager is the key manager for drive encryption and will permit drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes—for example, backup sets, export volumes, and database backup volumes—will not be encrypted.) If you specify ON and you enable either the library or system method of encryption, drive encryption will not be permitted and backup operations will fail.

**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is permitted if either the library or system method of encryption is enabled.

**OFF**

Specifies that drive encryption will not be permitted. If you enable either the library or system method of encryption, backups will fail. If you enable the application method, Tivoli Storage Manager will disable encryption and backups will be attempted.

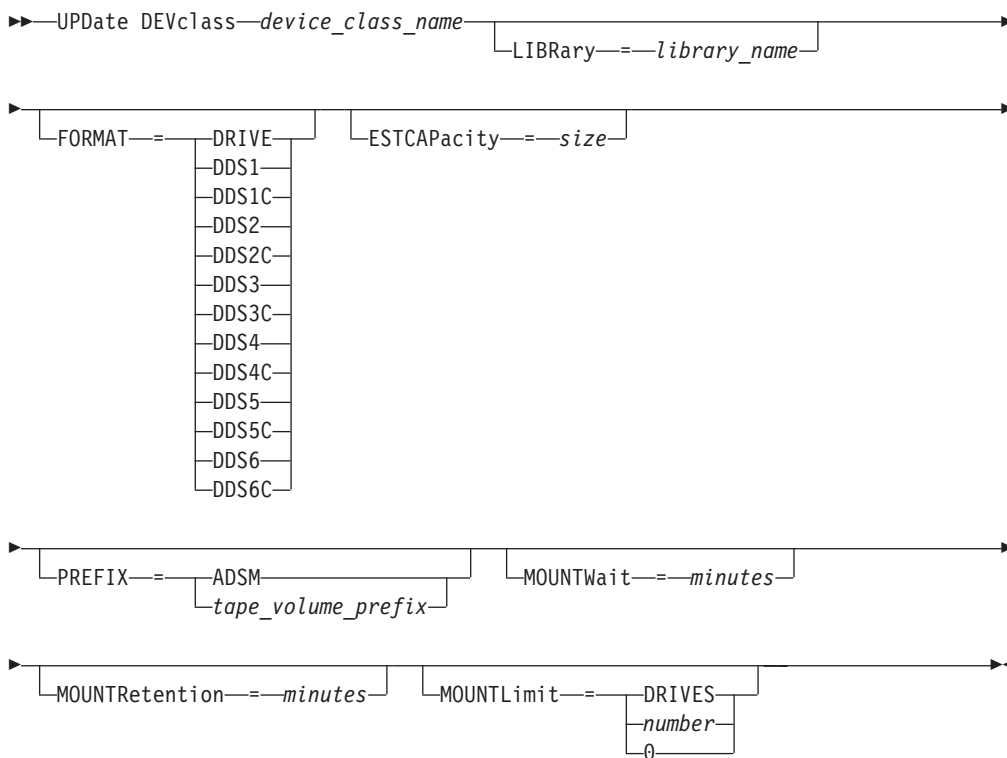
### UPDATE DEVCLASS (Update a 4MM device class)

Use the 4MM device class when you are using 4 mm tape devices.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined.

##### LIBRARY

Specifies the name of the defined library object that contains the 4 mm tape drives used by this device class. This parameter is optional. For information about defining a library object, see the DEFINE LIBRARY command.

##### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for 4 mm devices:

Table 353. Recording formats and default estimated capacities for 4 mm tapes

| Format | Estimated Capacity                                 | Description  |
|--------|--|--|
| DRIVE  | —  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| DDS1   | 1.3 GB (60-meter)<br>2.0 GB (90-meter)             | Uncompressed format, applies only to 60-meter and 90-meter tapes   |
| DDS1C  | See note<br>1.3 GB (60-meter)<br>2.0 GB (90-meter) | Compressed format, applies only to 60-meter and 90-meter tapes   |
| DDS2   | 4.0 GB   | Uncompressed format, only applies to 120-meter tapes   |
| DDS2C  | See note<br>8.0 GB                                 | Compressed format, only applies to 120-meter tapes   |
| DDS3   | 12.0 GB  | Uncompressed format, only applies to 125-meter tapes   |
| DDS3C  | See note<br>24.0 GB                                | Compressed format, only applies to 125-meter tapes   |
| DDS4   | 20.0 GB  | Uncompressed format, only applies to 150-meter tapes   |
| DDS4C  | See note<br>40.0 GB                                | Compressed format, only applies to 150-meter tapes   |
| DDS5   | 36 GB  | Uncompressed format, when using DAT 72 media   |
| DDS5C  | See note<br>72 GB                                  | Compressed format, when using DAT 72 media   |
| DDS6   | 80 GB  | Uncompressed format, when using DAT 160 media  |
| DDS6C  | See note<br>160 GB                                 | Compressed format, when using DAT 160 media  |

**Note:** If 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.

### ESTCAPacity

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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for 4 mm tapes, see Table 353 on page 999.

**PREFIX**

Specifies the high level qualifier of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional. The maximum length of this prefix is 8 characters.

If you have already established a tape label naming convention that supports your current tape management system, consider using a tape volume prefix that conforms to your naming conventions.

**MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true mount limit value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

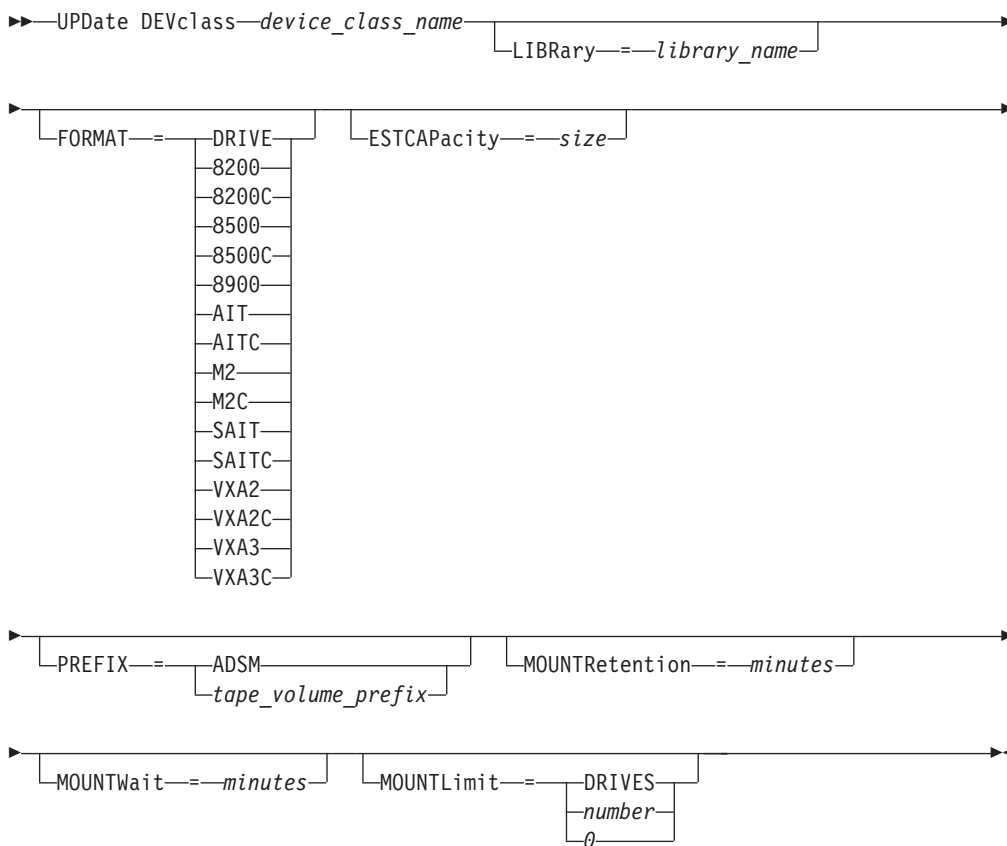
Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

**UPDATE DEVCLASS (Update an 8MM device class)**

Use the 8MM device class when you are using 8 mm tape devices.

**Privilege class**

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

**Syntax****Parameters*****device\_class\_name* (Required)**

Specifies the name of the device class to be updated.

**LIBRARY**

Specifies the name of the defined library object that contains the 8 mm tape drives that can be used by this device class. For information about defining a library object see the DEFINE LIBRARY command.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for 8 mm devices:



Table 354. Recording format and default estimated capacity for 8 mm tape

| Format      |                              |  |
|-------------|------------------------------|--|
| Medium Type | Estimated Capacity           | Description  |
| DRIVE       | –                            | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 8200        | 2.3 GB                       | Uncompressed (standard) format, using standard 112-meter tape cartridges   |
| 8200C       | See note<br>3.5 GB<br>4.6 GB | Compressed format, using standard 112-meter tape cartridges  |
| 8500        | See note                     | Drives (Read Write)  |
| 15m         | 600 MB                       | Eliaint 820 (RW)   |
| 15m         | 600 MB                       | Exabyte 8500/8500C (RW)  |
| 15m         | 600 MB                       | Exabyte 8505 (RW)  |
| 54m         | 2.35 GB                      | Eliaint 820 (RW)   |
| 54m         | 2.35 GB                      | Exabyte 8500/8500C (RW)  |
| 54m         | 2.35 GB                      | Exabyte 8505 (RW)  |
| 112m        | 5 GB or 10.0 GB              | Eliaint 820 (RW)   |
| 112m        | 5 GB or 10.0 GB              | Exabyte 8500/8500C (RW)  |
| 112m        | 5 GB or 10.0 GB              | Exabyte 8505 (RW)  |
| 160m XL     | 7 GB                         | Eliaint 820 (RW)   |
| 8500C       | See note                     | Drives (Read Write)  |
| 15m         | 1.2 GB                       | Eliaint 820 (RW)   |
| 15m         | 1.2 GB                       | Exabyte 8500/8500C (RW)  |
| 15m         | 1.2 GB                       | Exabyte 8505 (RW)  |
| 54m         | 4.7 GB                       | Eliaint 820 (RW)   |
| 54m         | 4.7 GB                       | Exabyte 8500/8500C (RW)  |
| 54m         | 4.7 GB                       | Exabyte 8505 (RW)  |
| 112m        | 5 GB or 10.0 GB              | Eliaint 820 (RW)   |
| 112m        | 5 GB or 10.0 GB              | Exabyte 8500/8500C (RW)  |
| 112m        | 5 GB or 10.0 GB              | Exabyte 8505 (RW)  |
| 160m XL     | 7 GB                         | Eliaint 820 (RW)   |
| 8900        | See note                     | Drive (Read Write)   |
| 15m         | –                            | Mammoth 8900 (R)   |
| 54m         | –                            | Mammoth 8900 (R)   |
| 112m        | –                            | Mammoth 8900 (R)   |
| 160m XL     | –                            | Mammoth 8900 (R)   |
| 22m         | 2.5 GB                       | Mammoth 8900 (RW)  |
| 125m        | –                            | Mammoth 8900 (RW with upgrade)   |
| 170m        | 40 GB                        | Mammoth 8900 (RW)  |

Table 354. Recording format and default estimated capacity for 8 mm tape (continued)

| Format      |                    |                                |
|-------------|--------------------|--------------------------------|
| Medium Type | Estimated Capacity | Description                    |
| AIT         | See note           | Drive                          |
| SDX1-25C    | 25 GB              | AIT, AIT2 and AIT3 drives      |
| SDX1-35C    | 35 GB              | AIT, AIT2 and AIT3 drives      |
| SDX2-36C    | 36 GB              | AIT2 and AIT3 drives           |
| SDX2-50C    | 50 GB              | AIT2 and AIT3 drives           |
| SDX3-100C   | 100 GB             | AIT3, AIT4, and AIT5 drives    |
| SDX3X-150C  | 150 GB             | AIT3-Ex, AIT4, and AIT5 drives |
| SDX4-200C   | 200 GB             | AIT4 and AIT5 drives           |
| SDX5-400C   | 400 GB             | AIT5 drive                     |
| AITC        | See note           | Drive                          |
| SDX1-25C    | 50 GB              | AIT, AIT2 and AIT3 drives      |
| SDX1-35C    | 91 GB              | AIT, AIT2 and AIT3 drives      |
| SDX2-36C    | 72 GB              | AIT2 and AIT3 drives           |
| SDX2-50C    | 130 GB             | AIT2 and AIT3 drives           |
| SDX3-100C   | 260 GB             | AIT3, AIT4, and AIT5 drives    |
| SDX3X-150C  | 390 GB             | AIT3-Ex, AIT4, and AIT5 drives |
| SDX4-200C   | 520 GB             | AIT4 and AIT5 drives           |
| SDX5-400C   | 1040 GB            | AIT5 drive                     |
| M2          | See note           | Drive (Read Write)             |
| 75m         | 20.0 GB            | Mammoth II (RW)                |
| 150m        | 40.0 GB            | Mammoth II (RW)                |
| 225m        | 60.0 GB            | Mammoth II (RW)                |
| M2C         | See note           | Drive (Read Write)             |
| 75m         | 50.0 GB            | Mammoth II (RW)                |
| 150m        | 100.0 GB           | Mammoth II (RW)                |
| 225m        | 150.0 GB           | Mammoth II (RW)                |
| SAIT        | See note           | Drive (Read Write)             |
|             | 500 GB             | Sony SAIT1-500(RW)             |
| SAITC       | See note           | Drive (Read Write)             |
|             | 1300 GB (1.3 TB)   | Sony SAIT1-500(RW)             |
| VXA2        | See note           | Drive (Read Write)             |
| V6 (62m)    | 20 GB              | VXA-2                          |
| V10 (124m)  | 40 GB              |                                |
| V17 (170m)  | 60 GB              |                                |
| VXA2C       | See note           | Drive (Read Write)             |
| V6 (62m)    | 40 GB              | VXA-2                          |
| V10 (124m)  | 80 GB              |                                |
| V17 (170m)  | 120 GB             |                                |
| VXA3        | See note           | Drive (Read Write)             |
| X6 (62m)    | 40 GB              | VXA-3                          |
| X10 (124m)  | 86 GB              |                                |
| X23 (230m)  | 160 GB             |                                |

Table 354. Recording format and default estimated capacity for 8 mm tape (continued)

| Format      |                    |                    |
|-------------|--------------------|--------------------|
| Medium Type | Estimated Capacity | Description        |
| VXA3C       | See note           | Drive (Read Write) |
| X6 (62m)    | 80 GB              | VXA-3              |
| X10 (124m)  | 172 GB             |                    |
| X23 (230m)  | 320 GB             |                    |

**Note:** The actual capacities may vary depending on which cartridges and drives are used.

- For the AITC and SAITC formats, the normal compression ratio is 2.6:1.
- For the M2C format, the normal compression ratio is 2.5:1.

### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to this device class. This parameter is optional.

You must specify this value as an integer followed by **K** (kilobytes), **M** (megabytes), or **G** (gigabytes).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB. The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For more information on the default estimated capacity for 8 mm tapes, see Table 354 on page 1003.

### PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional. The maximum length of this prefix is 8 characters.

If you have already established a tape label naming convention that supports your current tape management system, consider using a tape volume prefix that conforms to your naming conventions.

### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

For EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### MOUNTLimit

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### 0 (zero)

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete, but new transactions will be terminated.

### Example: Update the mount limit and capacity of an 8 mm device class

Update a device class named 8MMTAPE. Change the mount limit to 3 and the estimated capacity to 10 GB.

```
update devclass 8mmtape mountlimit=3 estcapacity=10G
```

### Example: Update the mount retention period of an 8 mm device class

Update an 8 mm device class named 8MMTAPE to a 15 minute mount retention.

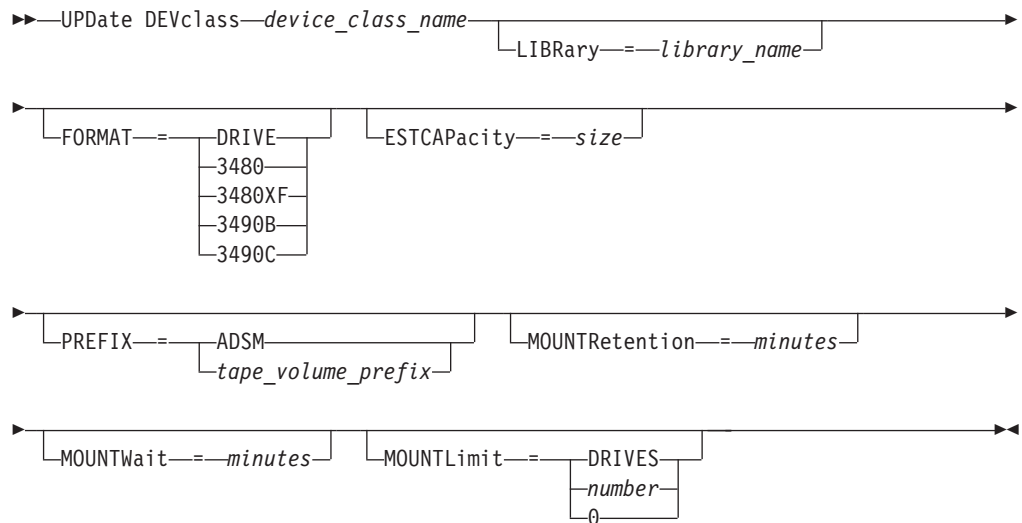
```
update devclass 8mmtape mountretention=15
```

**UPDATE DEVCLASS (Update a cartridge device class)**

Use the CARTRIDGE device class when you are using IBM 3480 and 3490 cartridge tape devices.

**Privilege class**

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

**Syntax****Parameters*****device\_class\_name* (Required)**

Specifies the name of the device class to be updated.

**LIBRARY**

Specifies the name of the defined library object that contains the CARTRIDGE tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for Cartridge System Tapes (CST):

Table 355. Recording format and default estimated capacity for cartridge tape volumes

| Format | Estimated capacity | Description   |
|--------|--------------------|---|
| DRIVE  | —                  | Specifies that the server selects the highest format that can be supported by the sequential access drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying the DRIVE value when a mixture of devices is used within the same library.<br><br>For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 3480   | 180 MB             | 18-track basic recording format (CST)   |
| 3480XF | 180 MB             | 18-track compressed recording format (CST)  |
| 3490B  | See note           | 36-track  |
|        | 360 MB<br>720 MB   | Uncompressed (basic) recording format (CST)<br>Compressed recording format (ECCST)  |
| 3490C  | See note           | 36-track  |
|        | 360 MB<br>720 MB   | Uncompressed (basic) recording format (CST)<br>Compressed recording format (ECCST)  |

**Note:** The actual capacities may vary depending on which cartridges and drives are used.

**ESTCAPacity**

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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for cartridge tapes, see Table 355.

**PREFIX**

Specifies the high-level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional.

If you have already established a tape label naming convention that supports your current tape management system, consider using a tape volume prefix that conforms to your naming conventions.

**MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

#### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

#### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true mountlimit value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

### **Example: Update the mount retention period for a device class**

Update the amount of time to retain idle mounted volumes for the SQUARE1 device class to 75 minutes.

```
update devclass square1 mountretention=75
```





read, write, delete, purge, and query data on a Centera storage device. To provide tighter control, create a PEA file using the command line interface provided by EMC Centera. For details about Centera authentication and authorization, refer to the EMC Centera *Programmer's Guide*.

2. You can also specify the PEA file name and path in an environment variable using the syntax `CENTERA_PEA_LOCATION=filePath_fileName`. The PEA file name and path specified using this environment variable apply to all Centera clusters. If you use this variable, you do not need to specify the PEA file name and path using the `HLADDRESS` parameter.
3. Updating the device class with a new or changed PEA file name and location could require a server restart if the Centera storage device identified by the IP address has already been accessed in the current instance of the Tivoli Storage Manager server.

### **MINCAPacity**

Specifies the new minimum size for Centera volumes assigned to a storage pool in this device class. This value represents the minimum amount of data stored on a Centera volume before the Tivoli Storage Manager server marks it full. Centera volumes will continue to accept data until the minimum amount of data has been stored. This parameter is optional.

*size*

Specify this value as an integer followed by **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The minimum value allowed is 1 MB (`MINCAPACITY=1M`). The maximum value allowed is 128 GB (`MINCAPacity=128G`).

### **MOUNTLimit**

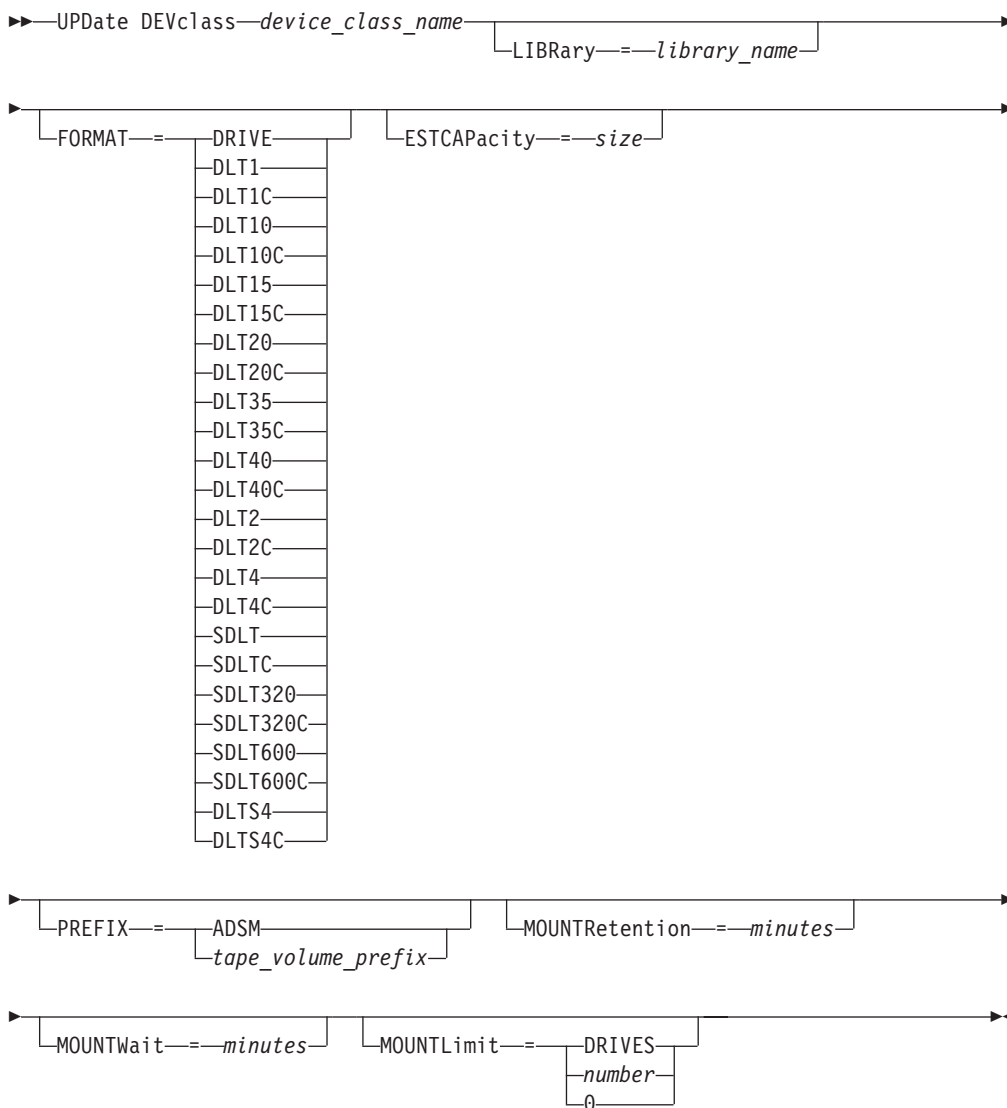
Specifies the new maximum number of sessions accessing the Centera device. This parameter is optional. You can specify any number from 0 or greater; however, the sum of all mount limit values for all device classes assigned to the same Centera device should not exceed the maximum number of sessions allowed by Centera.

**UPDATE DEVCLASS (Update a DLT device class)**

Use the DLT device class when you are using DLT tape devices.

**Privilege class**

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

**Syntax****Parameters****device\_class\_name (Required)**

Specifies the name of the device class to be updated.

**LIBRARY**

Specifies the name of the defined library object that contains the DLT tape drives that can be used by this device class. For information about defining a library object see the DEFINE LIBRARY command.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for DLT devices:

*Table 356. Recording format and default estimated capacity for DLT*

| Format | Estimated Capacity                  | Description  |
|--------|-------------------------------------|--|
| DRIVE  | –                                   | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| DLT1   | 40.0 GB                             | Uncompressed format, using only CompacTape III or CompacTape IV cartridges   |
| DLT1C  | See note 1 on page 1015.<br>80.0 GB | Compressed format, using only CompacTape III and CompacTape IV cartridges  |
| DLT10  | 10.0 GB                             | Uncompressed format, using only CompacTape III or CompacTape IV cartridges   |
| DLT10C | See note 1 on page 1015.<br>20.0 GB | Compressed format, using only CompacTape III and CompacTape IV cartridges  |
| DLT15  | 15.0 GB                             | Uncompressed format, using only CompacTape IIIxt or CompacTape IV cartridges (not CompacTape III)<br><b>Note:</b> Valid with DLT2000XT, DLT4000, and DLT7000 drives  |
| DLT15C | See note 1 on page 1015.<br>30.0 GB | Compressed format, using only CompacTape IIIxt or CompacTape IV cartridges (not CompacTape III)<br>Valid with DLT2000XT, DLT4000, and DLT7000 drives   |
| DLT20  | 20.0 GB                             | Uncompressed format, using only CompacTape IV cartridges<br>Valid with DLT4000, DLT7000, and DLT8000 drives  |
| DLT20C | See note 1 on page 1015.<br>40.0 GB | Compressed format, using only CompacTape IV cartridges<br>Valid with DLT4000, DLT7000, and DLT8000 drives  |
| DLT35  | 35.0 GB                             | Uncompressed format, using only CompacTape IV cartridges<br>Valid with DLT7000 and DLT8000 drives  |
| DLT35C | See note 1 on page 1015.<br>70.0 GB | Compressed format, using only CompacTape IV cartridges<br>Valid with DLT7000 and DLT8000 drives  |

Table 356. Recording format and default estimated capacity for DLT (continued)

| Format                              | Estimated Capacity                   | Description   |
|-------------------------------------|--------------------------------------|---|
| DLT40                               | 40.0 GB                              | Uncompressed format, using CompacTape IV cartridges<br>Valid with a DLT8000 drive             |
| DLT40C                              | See note 1 on page 1015.<br>80.0 GB  | Compressed format, using CompacTape IV cartridges<br>Valid with a DLT8000 drive               |
| DLT2                                | 80.0 GB                              | Uncompressed format, using Quantum DLT tape VS1 media   |
| DLT2C                               | See note 1 on page 1015.<br>160.0 GB | Compressed format, using Quantum DLT tape VS1 media   |
| DLT4                                | 160.0 GB                             | Uncompressed format, using Quantum DLTtape VS1 cartridges.<br>Valid with Quantum DLT-V4 drive |
| DLT4C                               | See note 1 on page 1015.<br>320.0 GB | Compressed format, using Quantum DLTtape VS1 cartridges.<br>Valid with Quantum DLT-V4 drive   |
| SDLT<br>See note 2 on page 171.     | 100.0 GB                             | Uncompressed format, using Super DLT Tape 1 cartridges<br>Valid with a Super DLT drive        |
| SDLTC<br>See note 2 on page 171.    | See note 1 on page 171.<br>200.0 GB  | Compressed format, using Super DLT Tape 1 cartridges<br>Valid with a Super DLT drive          |
| SDLT320<br>See note 2 on page 171.  | 160.0 GB                             | Uncompressed format, using Quantum SDLT I media<br>Valid with a Super DLT drive               |
| SDLT320C<br>See note 2 on page 171. | See note 1 on page 171.<br>320.0 GB  | Compressed format, using Quantum SDLT I media<br>Valid with a Super DLT drive                 |
| SDLT600                             | 300.0 GB                             | Uncompressed format, using SuperDLTtape-II media<br>Valid with a Super DLT drive              |
| SDLT600C                            | See note 1 on page 1015.<br>600.0 GB | Compressed format, using SuperDLTtape-II media<br>Valid with a Super DLT drive                |
| DLTS4                               | 800 GB                               | Uncompressed format, using Quantum DLT S4 media.<br>Valid with a DLT-S4 drive                 |
| DLTS4C                              | See note 1 on page 1015.<br>1.6 TB   | Compressed format, using Quantum DLT S4 media.<br>Valid with a DLT-S4 drive                   |

Table 356. Recording format and default estimated capacity for DLT (continued)

| Format | Estimated Capacity | Description |
|--------|--------------------|-------------|
|--------|--------------------|-------------|

**Note:**

1. Depending on the effectiveness of compression, the actual capacity may be greater than the listed value.
2. IBM Tivoli Storage Manager does not support a library that contains both Backward Read Compatible (BRC) SDLT and Non-Backward Read Compatible (NBRC) SDLT drives.

**ESTCAPacity**

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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB. The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For more information on estimated capacities, see Table 356 on page 1013.

**PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

**MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### 0 (zero)

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete, but new transactions will be terminated.

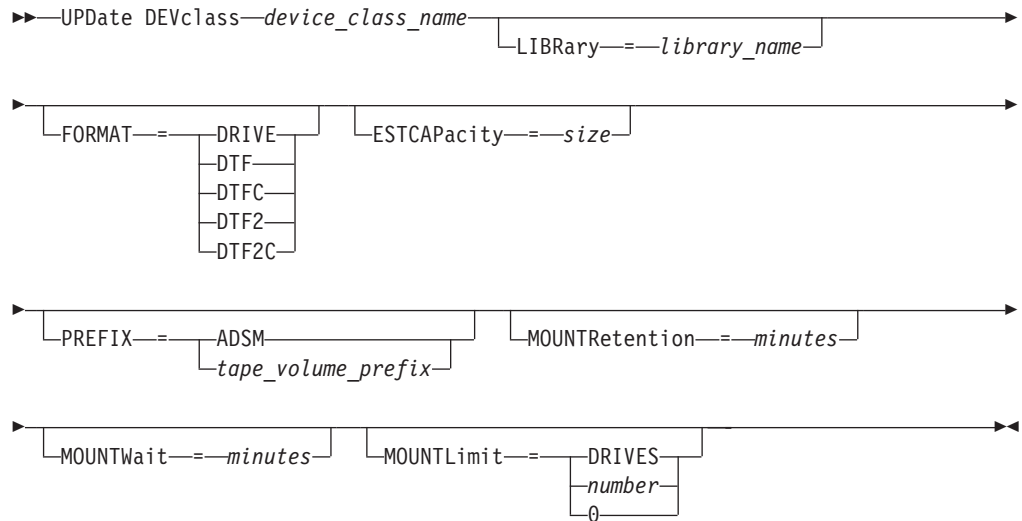
## UPDATE DEVCLASS (Update a DTF device class)

Use the DTF device class when you are using DTF tape devices.

### Privilege class

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

### Syntax



### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

#### LIBRARY

Specifies the name of the defined library object that contains the DTF tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

#### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for DTF devices:

Table 357. Recording format and default estimated capacity for DTF

| Format | Estimated Capacity   | Description  |
|--------|----------------------|--|
| DRIVE  | –                    | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| DTF    | 12.0 GB<br>42.0 GB   | Using GW-240 tape cassettes<br>Using GW-730L tape cassettes  |
| DTFC   | 24.0 GB<br>84.0 GB   | Using GW-240 tape cassettes<br>Using GW-730L tape cassettes  |
| DTF2   | 60.0 GB<br>200.0 GB  | Using GW2-60GS tape cassettes<br>Using GW2-200GL tape cassettes  |
| DTF2C  | 120.0 GB<br>400.0 GB | Using GW2-60GS tape cassettes<br>Using GW2-200GL tape cassettes  |

**ESTCAPacity**

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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For more information on estimated capacities, see Table 357.

**PREFIX**

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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, use a volume prefix that conforms to your naming conventions.

**MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true mountlimit value, (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

## UPDATE DEVCLASS — ECARTRIDGE

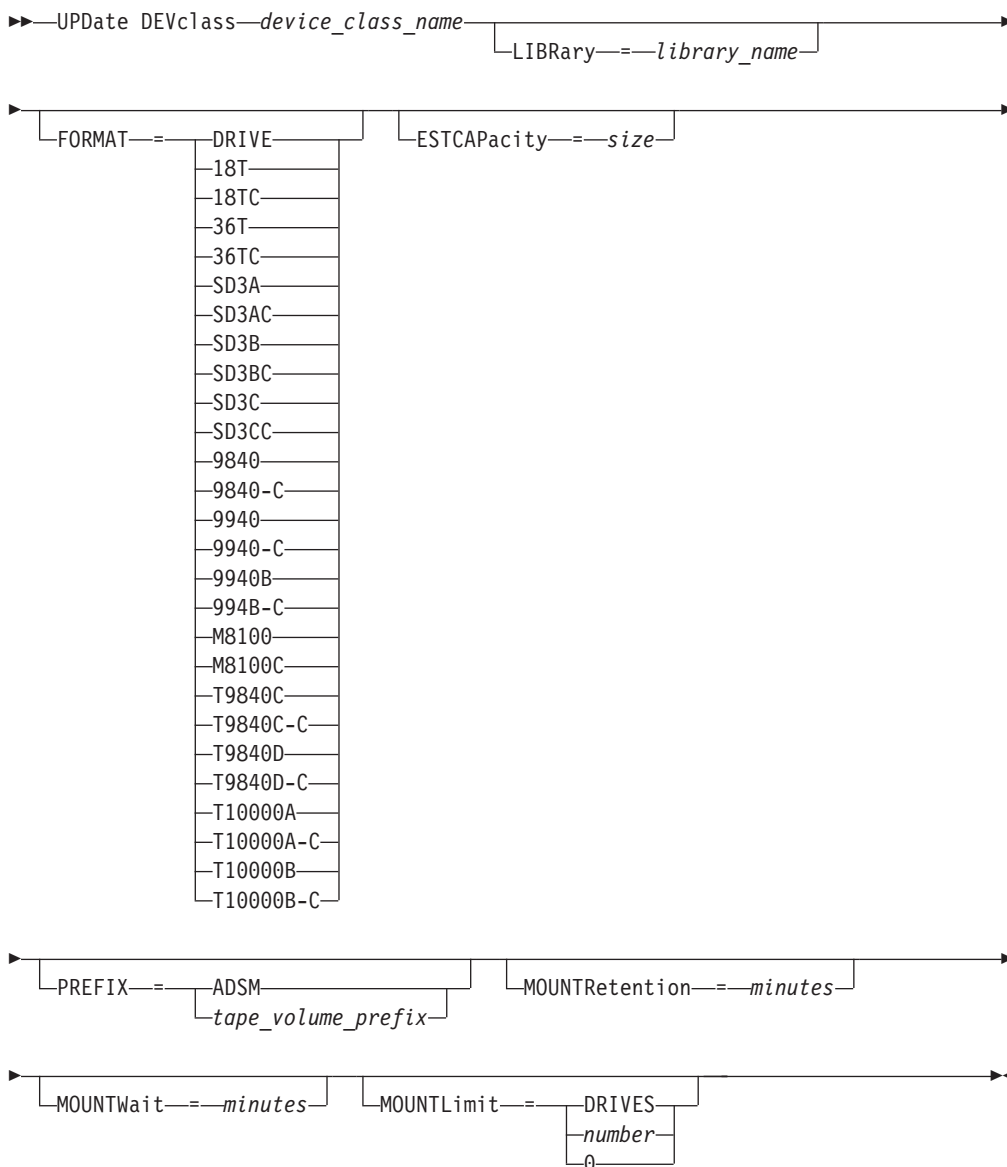
### UPDATE DEVCLASS (Update an ECARTRIDGE device class)

Use the ECARTRIDGE device class when you are using StorageTek drives such as the StorageTek SD-3, 9490, 9840 or 9940.

#### Privilege class

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

#### Syntax



#### Parameters

##### `device_class_name` (Required)

Specifies the name of the device class to be updated.

##### `LIBRary`

Specifies the name of the defined library object with the ECARTRIDGE tape

drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

### FORMAT

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

**Attention:** If you specify DRIVE for a device class that has non-compatible sequential access devices, then you must mount volumes on devices that are capable of reading or writing the format established when the volume was first mounted. This can cause delays if the only sequential access device that can access the volume is already in use.

The following table lists the recording formats and estimated capacities for ECARTRIDGE devices:

*Table 358. Recording formats and default estimated capacities for ECARTRIDGE tapes*

| Format | Estimated Capacity | Description  |
|--------|--------------------|--|
| DRIVE  | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 18T    | 360 MB             | 18-track uncompressed (standard) (read-only) format.   |
| 18TC   | 1.44 GB            | 18-track extended (read-only), compressed format.  |
| 36T    | 720 MB             | 36-track extended (read and write) format.   |
| 36TC   | 2.88 GB            | 36-track extended (read and write), compressed format.   |
| SD3A   | 10 GB              | Uncompressed (standard) format, using a 10 GB 'A' cartridge with 91 meters (298 feet) of tape.   |
| SD3AC  | 40 GB              | Compressed format, using a 10 GB 'A' cartridge with 91 meters (298 feet) of tape.  |
| SD3B   | 25 GB              | Uncompressed (standard) format, using a 25 GB 'B' cartridge with 204 meters (668 feet) of tape.  |
| SD3BC  | 100 GB             | Compressed format, using a 25 GB 'B' cartridge with 204 meters (668 feet) of tape.   |
| SD3C   | 50 GB              | Uncompressed (standard) format, using a 50 GB 'C' cartridge with 392 meters (1286 feet) of tape.   |
| SD3CC  | 200 GB             | Compressed format, using a 50 GB 'C' cartridge with 392 meters (1286 feet) of tape.  |
| 9840   | 20 GB              | Uncompressed 9840 format, using a Sun StorageTek 9840 cartridge  |
| 9840-C | 80 GB              | LZ-1 Enhanced (4:1) compressed 9840 format, using a Sun StorageTek 9840 cartridge  |
| 9940   | 60 GB              | Uncompressed 9940 format, using a Sun StorageTek 9940 cartridge  |
| 9940-C | 120 GB             | Compressed 9940 format, using a Sun StorageTek 9940 cartridge  |

Table 358. Recording formats and default estimated capacities for ECARTRIDGE tapes (continued)

| Format    | Estimated Capacity | Description   |
|-----------|--------------------|---|
| 9940B     | 200 GB             | Uncompressed 9940B format, using a Sun StorageTek 9940 cartridge  |
| 9940B-C   | 400 GB             | Compressed 9940B format, using a Sun StorageTek 9940 cartridge  |
| M8100     | 10 GB              | Uncompressed (standard) format, using a 10 GB cartridge.  |
| M8100C    | 10 GB              | Compressed format, using a 10 GB cartridge. Because there is no mechanism to determine the type of cartridge in an M8100 drive, the server assumes a 10 GB estimated uncompressed capacity. |
| T9840C    | 40 GB              | Uncompressed T9840C format, using a Sun StorageTek 9840 cartridge   |
| T9840C-C  | 80 GB              | Compressed T9840C format, using a Sun StorageTek 9840 cartridge   |
| T9840D    | 75 GB              | Uncompressed T9840D format, using a Sun StorageTek 9840 cartridge   |
| T9840D-C  | 150 GB             | Compressed T9840D format, using a Sun StorageTek 9840 cartridge   |
| T10000A   | 500 GB             | Uncompressed T10000A format, using a Sun StorageTek T10000 cartridge  |
| T10000A-C | 1 TB               | Compressed T10000A format, using a Sun StorageTek T10000 cartridge  |
| T10000B   | 1 TB               | Uncompressed T10000B format, using a Sun StorageTek T10000 cartridge  |
| T10000B-C | 2 TB               | Compressed T10000B format, using a Sun StorageTek T10000 cartridge  |

**Notes:**

- Some formats use a tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity may be double or more than the listed value.
- T10000A drives can read and write the T10000A format only. T10000B drives can read, but cannot write, the T10000A format.

**ESTCAPacity**

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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for cartridge tapes, see Table 358 on page 1021.

**PREFIX**

Specifies the high-level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional.

If you have already established a tape label naming convention that supports your current tape management system, consider using a tape volume prefix that conforms to your naming conventions.

**MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

## UPDATE DEVCLASS — ECARTRIDGE

### 0 (zero)

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

UPDATE DEVCLASS (Update a FILE device class)

Use the FILE device class when you are using files on magnetic disk storage as volumes that store data sequentially (as on tape).

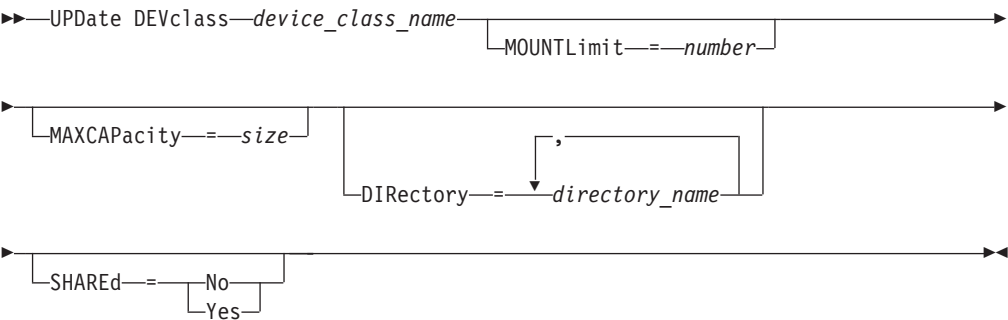
For information about disk subsystem requirements for FILE-type disk storage, see the *Administrator's Guide*.

The FILE device class does not support EXTERNAL libraries.

Privilege class

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

Syntax



Parameters

device\_class\_name (Required)

Specifies the name of the device class to be updated.

MOUNTLimit

Specifies the maximum number of files that can be simultaneously open for input and output. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

MAXCAPacity

Specifies the maximum size of any data storage files categorized by this device class. This parameter is optional.

Specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The minimum value allowed is 100 KB (MAXCAPACITY=100K).

For example, MAXCAPACITY=5G specifies that the maximum capacity for a volume in this device class is 5 gigabytes. The value specified should be less than or equal to the maximum supported size of a file on the target file system.

Do not define a MAXCAPACITY value greater than 640M when this file is for REMOVABLEFILE CD support. A value less than a CD's usable space (650 MB) allows for a one-to-one match between files from the FILE device class and copies that are on CD.

### DIRectory

Specifies the directory location or locations of the files used in this device class. Enclose the entire list of directories within quotation marks, using commas to separate individual directory names. Special characters (for example, blank spaces) are permitted within directory names. For example, the directory list "abc def,xyz" contains two directories: abc def and xyz. This parameter is optional.

By specifying a directory name or names, you identify the locations 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 or names into their fully qualified forms, starting from the root directory.

**Important:** If you are using storage agents for shared access to FILE volumes, you need to use the DEFINE PATH command to define a path for each storage agent. The path definition includes the directory names used by the storage agent to access each directory.

Later, if the server needs to allocate a scratch volume, it creates a new file in one of these directories. (The server can choose any of the directories in which to create new scratch volumes.) 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, if you define a device class with a directory of tsmstor and the server needs a scratch volume in this device class to store export data, the file that the server creates might be named /tsmstor/00566497.exp.

**Tip:** If you specify multiple directories for a device class, ensure that the directories are associated with separate file systems. Space trigger functions and storage pool space calculations take into account the space remaining in each directory. If you specify multiple directories for a device class and the directories reside in the same file system, the server will calculate space by adding values representing the space remaining in each directory. These space calculations will be inaccurate. Rather than choosing a storage pool with sufficient space for an operation, the server might choose the wrong storage pool and run out of space prematurely. For space triggers, an inaccurate calculation might result in a failure to expand the space available in a storage pool. Failure to expand space in a storage pool is one of the conditions that can cause a trigger to become disabled. If a trigger is disabled because the space in a storage pool could not be expanded, you can re-enable the trigger by issuing the following command: update spacetrigger stg. No further changes are required to the space trigger.

**Restriction:** To modify a list of directories, you must replace the entire list.

### SHARED

Specifies that this FILE device class will be shared between the server and one or more storage agents. To prepare for sharing, a library will be automatically defined along with a number of drives corresponding to the MOUNTLIMIT associated with the device class. If the library and drives already exist and the MOUNTLIMIT is changed, drives will either be created to reach a new higher MOUNTLIMIT value or deleted to reach a new lower value.



## Storage agents using FILE volumes

You must ensure that storage agents can access newly created FILE volumes. To access FILE volumes, storage agents replace names from the directory list in the device-class definition with the names in the directory list for the associated path definition. The following illustrates the importance of matching device classes and paths to ensure that storage agents can access newly created FILE volumes.

Suppose you want to use these three directories for a FILE library:

```
/usr/tivoli1
/usr/tivoli2
/usr/tivoli3
```

1. You use the following command to set up a FILE library named CLASSA with one drive named CLASSA1 on SERVER1:

```
define devclass classa devtype=file
directory="/usr/tivoli1,/usr/tivoli2,/usr/tivoli3"
shared=yes mountlimit=1
```

2. You want the storage agent STA1 to be able to use the FILE library, so you define the following path for storage agent STA1:

```
• define path server1 stal srctype=server desttype=drive device=file
  directory="/usr/ibm1,/usr/ibm2,/usr/ibm3" library=classa
```

In this scenario, the storage agent, STA1, will replace the directory name /usr/tivoli1 with the directory name /usr/ibm1 to access FILE volumes that are in the /usr/tivoli1 directory on the server.

The following results occur:

- If file volume /usr/tivoli1/file1.dsm is created on SERVER1, and if the following command is issued,

```
update devclass classa directory="/usr/otherdir,/usr/tivoli2,
/usr/tivoli3"
```

SERVER1 will still be able to access file volume /usr/tivoli1/file1.dsm, but the storage agent STA1 will not be able to access it because a matching directory name in the PATH directory list no longer exists. If a directory name is not available in the directory list associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume will still be accessible from the Tivoli Storage Manager server for reading, failure of the storage agent to access the FILE volume can cause operations to be retried on a LAN-only path or to fail.

### Example: Update a FILE device class for sharing

Prepare a FILE device class (named PLAINFILES) for sharing with a Tivoli Storage Manager storage agent.

```
update devclass plainfiles shared=yes
```

### Example: Update the capacity of a FILE device class

Update a file device class named STORFILES to a maximum capacity of 25 MB.

```
update devclass storfiles maxcap=25m
```

### Example: Add a directory to a FILE device class

Update the FILE device class, CLASSA, by adding a directory, /usr/otherdir, to the directory list. The directories /opt/tivoli2 and /opt/tivoli3 were specified when the device class was first defined.

```
update devclass classa
  directory="/opt/tivoli2,/opt/tivoli3,/usr/otherdir"
```

## UPDATE DEVCLASS (Update a GENERICTAPE device class)

Use the GENERICTAPE device class for tape drives supported by operating system device drivers.

When using this device type, the server does not recognize either the type of device or the cartridge recording format. Because the server does not recognize the type of device, if an I/O error occurs, error information is less detailed compared to error information for a specific device type (for example, 8MM). When defining devices to the server, do not mix various types of devices within the same device type.

### Privilege class

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

### Syntax

```

►►—UPDate DEVclass—device_class_name—┐
└LIBRARY—=library_name┘
┐
└ESTCAPacity—=size┘└MOUNTRetention—=minutes┘
┐
└MOUNTWait—=minutes┘└MOUNTLimit—=DRIVES
└number┘
└0┘

```

### Parameters

#### *device\_class\_name* (Required)

Specifies the name of the device class to be updated.

#### LIBRARY

Specifies the name of the defined library object that contains the tape drives used by this device class. This parameter is optional. For information about defining a library object, see the DEFINE LIBRARY command.

#### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to 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.

Specify a capacity appropriate to the particular tape drive being used.

You must specify this value as an integer followed by **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

#### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

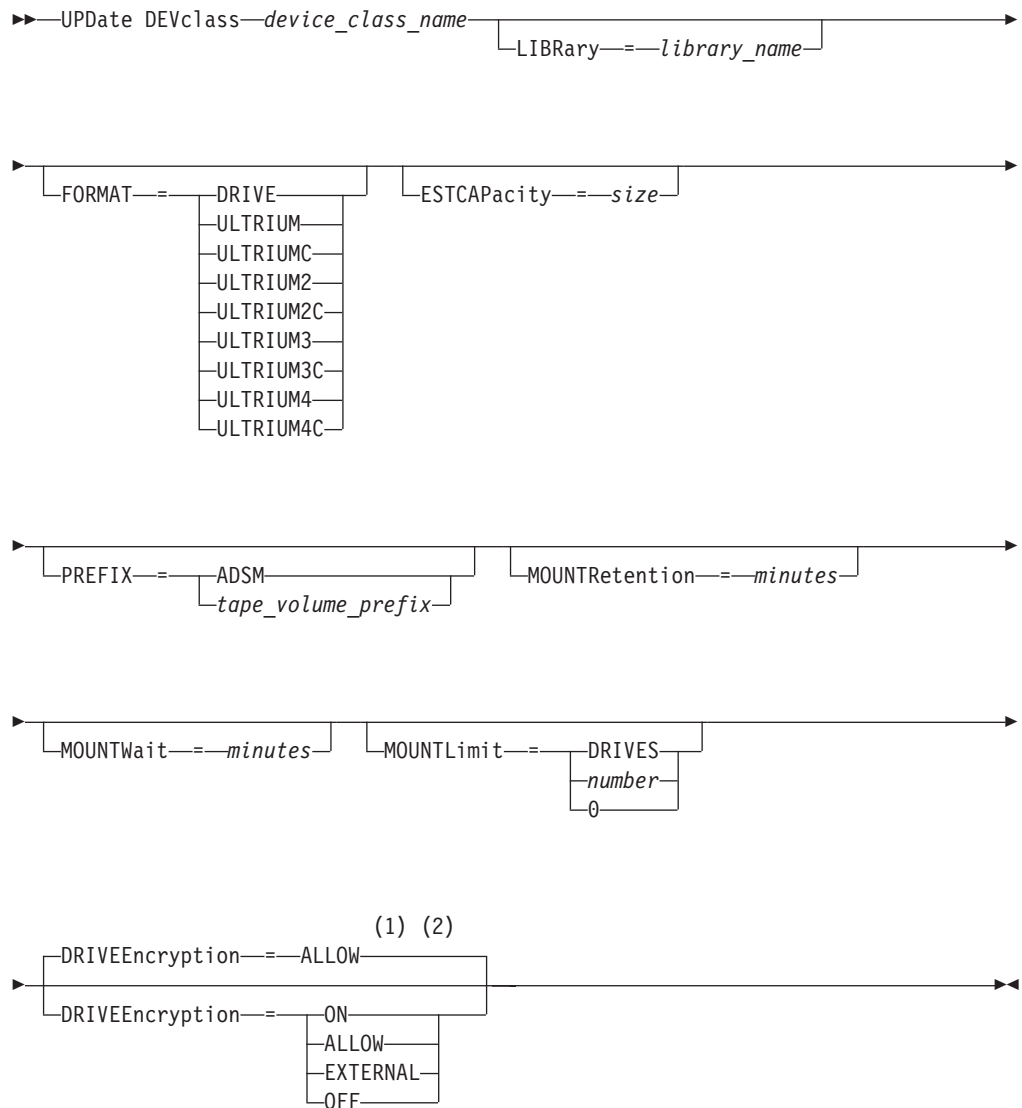
Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

**UPDATE DEVCLASS (Update an LTO device class)**

Use the LTO device class when you are using LTO tape devices.

**Privilege class**

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

**Syntax****Notes:**

- 1 You cannot specify DRIVEENCRYPTION=ON if your drives are using WORM (write once, read many) media.
- 2 Drive encryption is supported only for IBM and HP Ultrium 4 drives and media.

**Parameters***device\_class\_name* **(Required)**

Specifies the name of the device class to be updated. The maximum length of the device class name is 30 characters.

**LIBRARY**

Specifies the name of the defined library object that contains the LTO tape drives used by this device class. For information about defining a library object, see the **DEFINE LIBRARY** command.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

When migrating all drives from Ultrium to Ultrium 2 devices:

- Delete all existing Ultrium drive definitions and the paths associated with them.
- Define the new Ultrium 2 drives and paths.

If you are considering mixing different generations of LTO media and drives, be aware of the following restrictions. For more information about mixing drives and media, refer to the *Administrator's Guide*.

*Table 359. Read - write capabilities for different generations of LTO drives*

| Drives   | Generation 1 media | Generation 2 media | Generation 3 media | Generation 4 media |
|--|--------------------|--------------------|--------------------|--------------------|
| Generation 1   | Read and write     | n/a                | n/a                | n/a                |
| Generation 2   | Read and write     | Read and write     | n/a                | n/a                |
| Generation 3 <sup>1</sup>  | Read only          | Read and write     | Read and write     | n/a                |
| Generation 4 <sup>2</sup>  | n/a                | Read only          | Read and write     | Read and write     |
| <sup>1</sup> In a library with a Generation 3 drive, all Generation 1 scratch volumes need to be checked out, and all Generation 1 storage pool volumes need to be updated to read-only.<br><sup>2</sup> In a library with a Generation 4 drive, all Generation 2 scratch volumes need to be checked out, and all Generation 2 storage pool volumes need to be updated to read-only. |                    |                    |                    |                    |

The following table lists the recording formats and estimated capacities for LTO devices:

*Table 360. Recording format and default estimated capacity for LTO*

| Format   | Estimated capacity | Description   |
|----------|--------------------|---|
| DRIVE    | –                  | <p>The server selects the highest format that is supported by the drive on which a volume is mounted.</p> <p><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives.</p> |
| ULTRIUM  | 100 GB             | Uncompressed format, using Ultrium cartridges   |
| ULTRIUMC | See note<br>200 GB | Compressed format, using Ultrium cartridges   |

Table 360. Recording format and default estimated capacity for LTO (continued)

| Format    | Estimated capacity | Description  |
|-----------|--------------------|--|
| ULTRIUM2  | 200 GB             | Uncompressed (standard) format, using Ultrium 2 cartridges |
| ULTRIUM2C | See note<br>400 GB | Compressed format, using Ultrium 2 cartridges              |
| ULTRIUM3  | 400 GB             | Uncompressed (standard) format, using Ultrium 3 cartridges |
| ULTRIUM3C | See note<br>800 GB | Compressed format, using Ultrium 3 cartridges              |
| ULTRIUM4  | 800 GB             | Uncompressed (standard) format, using Ultrium 4 cartridges |
| ULTRIUM4C | See note<br>1.6 TB | Compressed format, using Ultrium 4 cartridges              |

**Note:** If 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.

### ESTCAPacity

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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=100G specifies that the estimated capacity for a volume in this device class is 100 GB.

For more information on estimated capacities, see Table 360 on page 1032.

### PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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, use a volume prefix that conforms to your naming conventions.

### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, for a library managed by an external media management program), you can enhance device sharing between applications by setting this parameter to a low value (for example, two minutes).

### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

### **DRIVEEncryption**

Specifies whether drive encryption will be permitted. This parameter is optional. The default is ALLOW.

**Note:** Drive encryption is supported only for IBM and HP Ultrium 4 drives and media.

### **ON**

Specifies that Tivoli Storage Manager is the key manager for drive encryption and will permit drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes will not be encrypted. For example, backup sets, export volumes, and database backup volumes are not encrypted.) If you specify ON and you enable another method of encryption, drive encryption will not be permitted and backup operations will fail.

**Note:** You cannot specify Tivoli Storage Manager as the key manager for drive encryption of WORM (write once, read many) media. (If you are using WORM media, you cannot specify DRIVEENCRYPTION=ON.)



**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is permitted if another method of encryption is enabled.

**EXTERNAL**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. Use this setting with an encryption methodology that is provided by another vendor and that is used with Application Method Encryption (AME) enabled on the drive. When you specify EXTERNAL and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager does not turn encryption off. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns encryption off.

**OFF**

Specifies that drive encryption will not be permitted. If you enable another method of encryption, backups will fail. If you enable the application method, Tivoli Storage Manager will disable encryption and backups will be attempted.

**Example: Update the mount limit for an LTO device class**

Update a device class named LTOTAPE. Change the mount limit to 2.

```
update devclass ltotape mountlimit=2
```

### UPDATE DEVCLASS (Update a NAS device class)

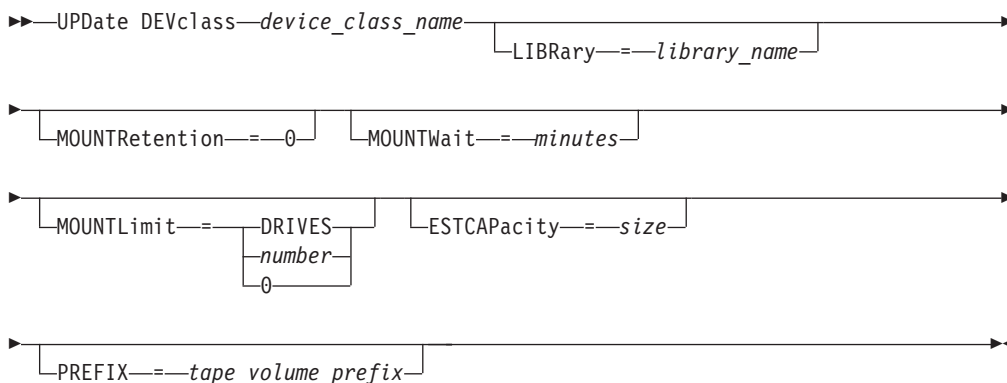
Use the NAS device class when you are using NDMP (Network Data Management Protocol) operations to back up network-attached storage (NAS) file servers. The device class is for drives supported by the NAS file server for backups.

The NAS device class does not support EXTERNAL libraries.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be defined. The maximum length of the device class name is 30 characters.

##### LIBRARY

Specifies the name of the defined library object that contains the SCSI tape drives used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.

##### MOUNTRetention=0

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. Zero (0) is the only supported value for device classes with DEVType=NAS.

##### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

##### MOUNTLimit

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a

simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### DRIVES

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

#### 0 (zero)

Specifies that no new transactions can gain access to the storage pool.

#### ESTCAPacity

Specifies the estimated capacity for the volumes assigned to this device class. This parameter is optional.

This value must be an integer followed by K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=100G specifies that the estimated capacity for a volume in this device class is 100 GB.

#### PREFIX

Specifies the high level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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, use a volume prefix that conforms to your naming conventions.

#### **Example: Update the estimated capacity for a NAS device class**

Update a device class named NASTAPE. Change the estimated capacity to 200 GB.

```
update devclass nastape library=naslib estcapacity=200G
```

## UPDATE DEVCLASS — OPTICAL and WORM types

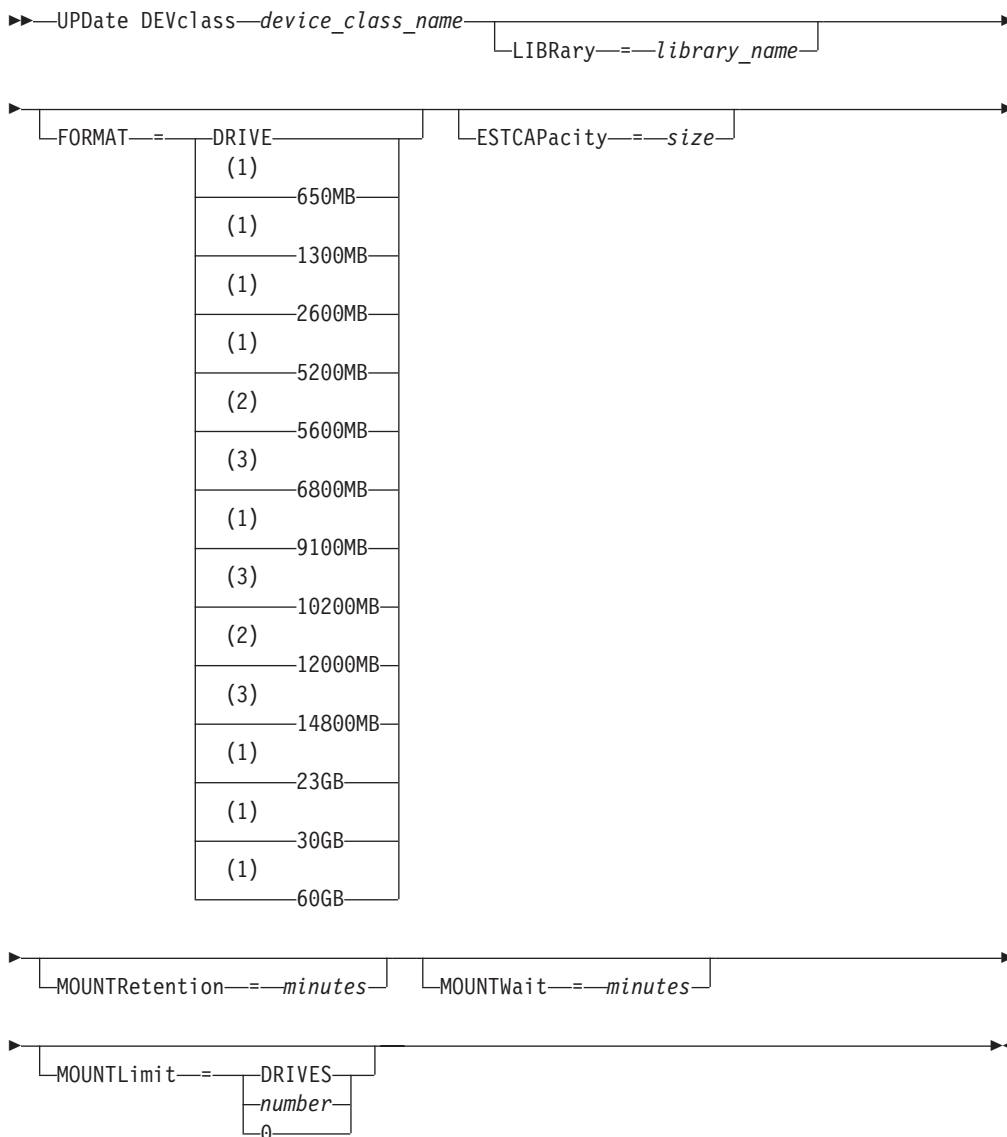
### UPDATE DEVCLASS (Update OPTICAL and WORM device classes)

Use the OPTICAL device class when you are using optical devices. Use the WORM device class when you are using optical devices that have WORM capability.

#### Privilege class

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

#### Syntax



#### Notes:

- 1 Available only on AIX and Windows. Not available for WORM12 or WORM14 device types
- 2 Available only for WORM12 device types
- 3 Available only for WORM14 device types

**Parameters***device\_class\_name* **(Required)**

Specifies the name of the device class to be defined.

**LIBRARY**

Specifies the name of the defined library object that contains the Optical or WORM drives that can be used by this device class. For information about defining a library object see the DEFINE LIBRARY command.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following table lists the recording formats and estimated capacities for OPTICAL and WORM devices:

*Table 361. Estimated capacities for OPTICAL and WORM drives*

| Format  | Estimated capacity | Description  |
|---------|--------------------|--|
| DRIVE   | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 650MB   | 650 MB             | Using a 650 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types  |
| 1300MB  | 1300 MB            | Using a 1300 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types   |
| 2600MB  | 2600 MB            | Using a 2600 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types   |
| 5200MB  | 5200 MB            | Using a 5200 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types   |
| 5600MB  | 5600 MB            | Using a 5600 MB 12-inch optical drive<br>Valid for WORM12 device types only  |
| 6800MB  | 6800 MB            | Using a 6800 MB 14-inch optical drive<br>Valid for WORM14 device types only  |
| 9100MB  | 9100 MB            | Using a 9100 MB 5.25-inch optical drive<br>Not valid for WORM12 or WORM14 device types   |
| 10200MB | 10200 MB           | Using a 10200 MB 14-inch optical drive<br>Valid for WORM14 device types only   |
| 12000MB | 12000 MB           | Using a 12000 MB 12-inch optical drive<br>Valid for WORM12 device types only   |

Table 361. Estimated capacities for OPTICAL and WORM drives (continued)

| Format  | Estimated capacity | Description  |
|---------|--------------------|--|
| 14800MB | 14800 MB           | Using a 14800 MB 14-inch optical drive<br>Valid for WORM14 device types only                       |
| 23GB    | 23 GB              | Using Sony Blue Laser read-write and WORM media<br>Not valid for WORM12 or WORM14 device types     |
| 30GB    | 30 GB              | Using Plasmon UDO1 read-write and WORM media<br>Not valid for WORM12 or WORM14 device types        |
| 60GB    | 60 GB              | Using Plasmon or IBM UDO2 read-write and WORM media<br>Not valid for WORM12 or WORM14 device types |

**Restriction:** If you are considering mixing different generations of UDO media and drives, be aware of the following:

- UDO1 drives can read and write UDO1 media only.
- UDO2 drives can read from, but not write to, UDO1 media.

### ESTCAPacity

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 K (kilobytes), M (megabytes), or G (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on estimated capacities, see Table 361 on page 1039.

### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types, setting this parameter to a low value (for example, two minutes) enhances device sharing between applications.

### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

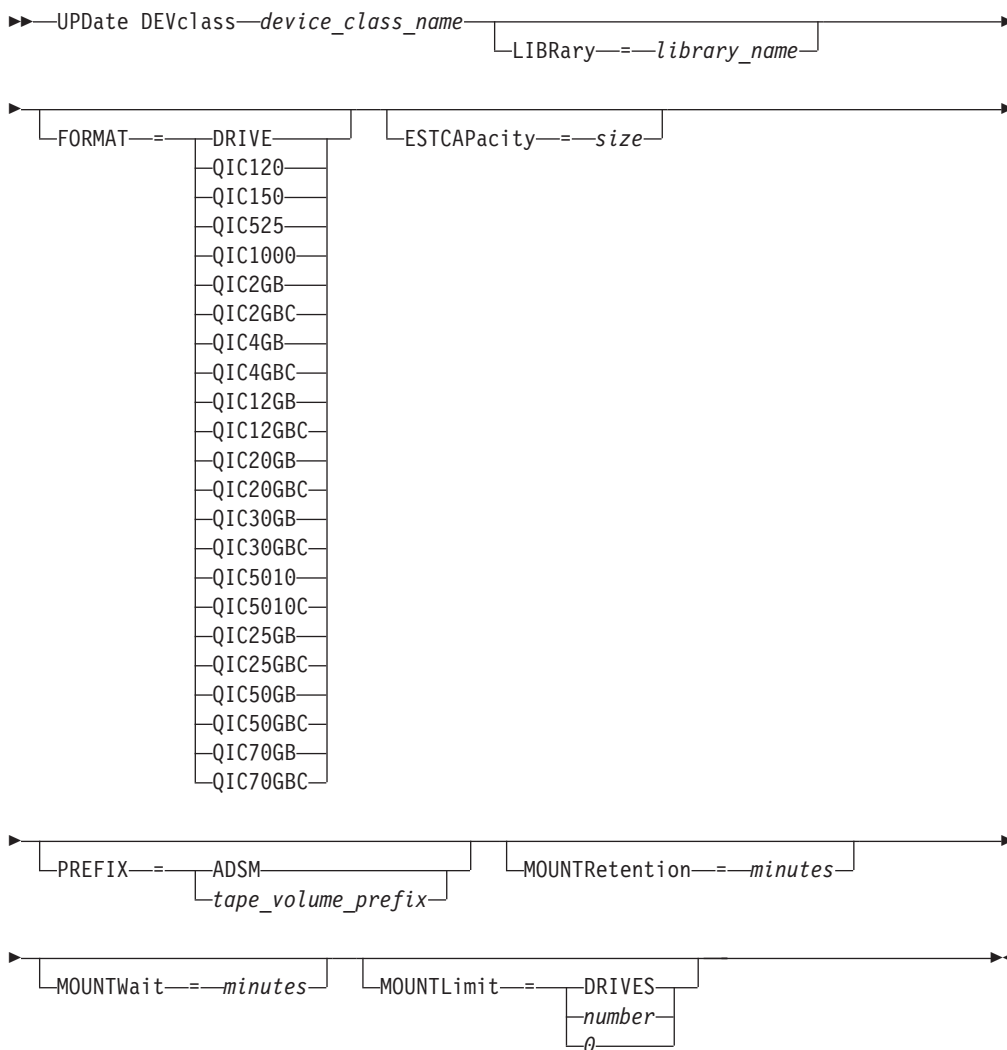
Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

**UPDATE DEVCLASS (Update a QIC device class)**

Use the QIC device class when you are using QIC tape devices.

**Privilege class**

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

**Syntax****Parameters****device\_class\_name (Required)**

Specifies the name of the device class to be defined.

**LIBRARY**

Specifies the name of the defined library object that contains the tape drives that can be used by this device class. For information about defining a library object, see the DEFINE LIBRARY command.



**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional.

The following tables list the recording formats, estimated capacities and recording format options for QIC devices:

*Table 362. Recording format and default estimated capacity for quarter-inch cartridge (QIC) tape*

| Format   | Estimated Capacity        | Description  |
|----------|---------------------------|--|
| DRIVE    | –                         | The server selects the highest format that can be supported by the sequential access drive on which a volume is mounted.<br><br>Avoid specifying the DRIVE value when a mixture of devices is used within the same library.<br><br>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   | 26 MB – 172 MB            | 120 QIC format, depending on media   |
| QIC150   | 31 MB – 207 MB            | 150 QIC format, depending on media   |
| QIC525   | 65 MB – 427 MB            | 525 QIC format, depending on media   |
| QIC1000  | 169 MB – 1.1 GB           | 1000 QIC format, depending on media  |
| QIC2GB   | 2 GB                      | Uncompressed 2000 QIC format   |
| QIC2GBC  | See note<br>4 GB          | Compressed 2000 QIC format   |
| QIC4GB   | 4 GB                      | Uncompressed 4000 QIC format   |
| QIC4GBC  | See note<br>8 GB          | Compressed 4000 QIC format   |
| QIC12GB  | 12 GB                     | Uncompressed 12000 QIC format, using 343-meter tape  |
| QIC12GBC | See note<br>24 GB         | Compressed 12000 QIC format, using 343-meter tape  |
| QIC5010  | 13 GB – 16 GB             | Uncompressed 5010 QIC format, depending on media   |
| QIC5010C | See note<br>26 GB – 32 GB | Compressed 5010 QIC format, depending on media   |
| QIC20GB  | 20 GB                     | Uncompressed 20000 QIC format  |
| QIC20GBC | See note<br>40 GB         | Compressed 20000 QIC format  |
| QIC25GB  | 25 GB                     | Uncompressed 25000 QIC format  |
| QIC25GBC | See note<br>50 GB         | Compressed 25000 QIC format  |
| QIC30GB  | 30 GB                     | Uncompressed 30000 QIC format  |

Table 362. Recording format and default estimated capacity for quarter-inch cartridge (QIC) tape (continued)

| Format   | Estimated Capacity | Description                  |
|----------|--------------------|------------------------------|
| QIC30GBC | See note<br>60 GB  | Compressed 30000 QIC format  |
| QIC50GB  | 50 GB              | Uncompressed 50GB QIC format |
| QIC50GBC | See note<br>100 GB | Compressed 50GB QIC format   |
| QIC70GB  | 70 GB              | Uncompressed 70GB QIC format |
| QIC70GBC | See note<br>140 GB | Compressed 70GB QIC format   |

**Note:** If 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.

Table 363. QIC tape recording format options

| 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:** The server cannot use 3M DC300XLP and 3M DC600A tapes.

### ESTCAPacity

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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for QIC tapes, see Table 362 on page 1043.

### PREFIX

Specifies the high level qualifier of the file name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set 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.

#### **MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

#### **MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

#### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

#### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

#### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

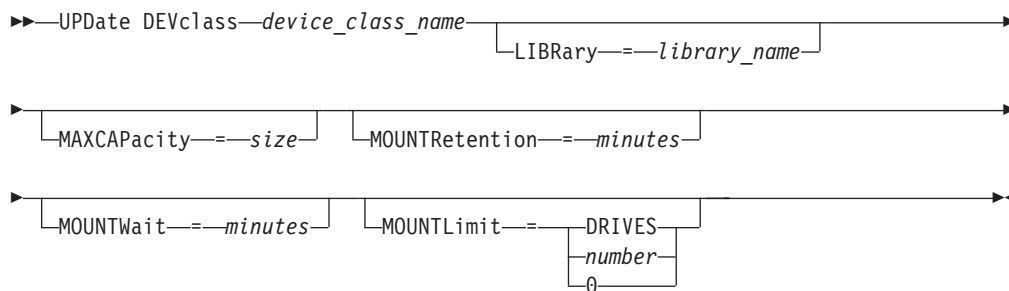
### UPDATE DEVCLASS (Update a REMOVABLEFILE device class)

Use the REMOVABLEFILE device class for removable media devices that are attached as local, removable file systems.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be updated.

##### LIBRARY

Specifies the name of the defined library object that contains the removable media drives used by this device class. This parameter is optional.

##### MAXCAPacity

Specifies the maximum size of any volumes defined to a storage pool categorized by this device class. This parameter is optional.

Because the server opens only one file per physical removable medium, specify a capacity that enables one file to make full use of your media capacity.

You must specify this value as an integer followed by K (kilobytes), M (megabytes), or G (gigabytes).

For example, `MAXCAPACITY=5M` specifies that the maximum capacity for a volume in this device class is 5 MB. The smallest value allowed is 100 KB (that is, `MAXCAPACITY=100K`).

##### MOUNTRetention

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

##### MOUNTWait

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true MOUNTLIMIT value (including online status).

**Note:** For EXTERNAL library types, do not specify DRIVES for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

*number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never exceed the number of drives that are defined and online in the library that services this device class.

**0 (zero)**

Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete but new transactions will be terminated.

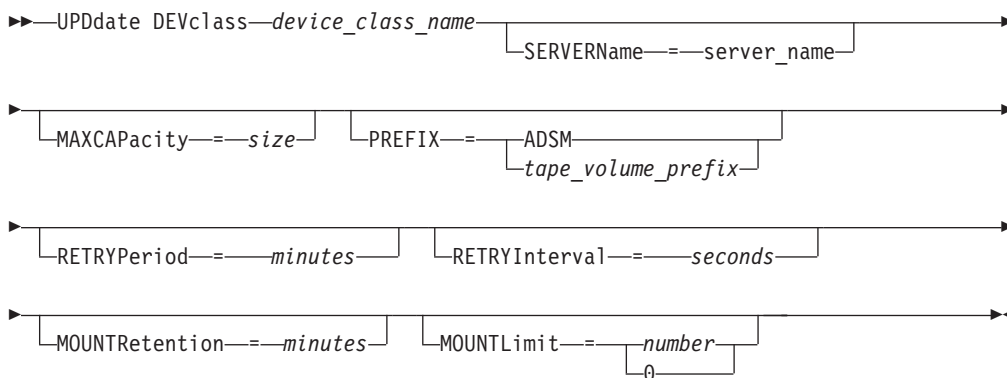
### UPDATE DEVCLASS (Update a SERVER device class)

Use the SERVER device class to use storage volumes or files archived in another Tivoli Storage Manager server.

#### Privilege class

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

#### Syntax



#### Parameters

##### *device\_class\_name* (Required)

Specifies the name of the device class to be updated.

##### SERVERName

Specifies the name of the server. The SERVERNAME parameter must match a defined server.

**Note:** If you change the SERVERNAME of an existing server to a new name, data on the volumes under the old SERVERNAME will no longer be accessible with this device class.

##### MAXCAPacity

Specifies the maximum size that objects can be when created on the target server. This parameter is optional.

Specify this value as an integer followed by **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The minimum value allowed is 100 KB (MAXCAPACITY=100K).

##### PREFIX

Specifies the beginning portion of the high-level archive file name on the target server. 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, use a volume prefix that conforms to your naming conventions.

##### RETRYPeriod

Specifies the retry period in minutes. The retry period is the interval during which the server attempts to contact a target server if there is a suspected communications failure. This parameter is optional.

**RETRYInterval**

Specifies the retry interval in seconds. The retry interval is how often retries are done within a given time period. This parameter is optional.

**MOUNTRetention**

Specifies the number of minutes to retain an idle connection with the target server before closing the connection. This parameter is optional. You can specify a number from 0 to 9999.

**MOUNTLimit**

Specifies the maximum number of simultaneous sessions between the source server and the target server. Any attempts to access more sessions than indicated by the mount limit cause the requester to wait. This parameter is optional. You can specify a number from 1 to 4096.

The following are possible values:

*number*

Specifies the maximum number of simultaneous sessions between the source server and the target server.

**0 (zero)**

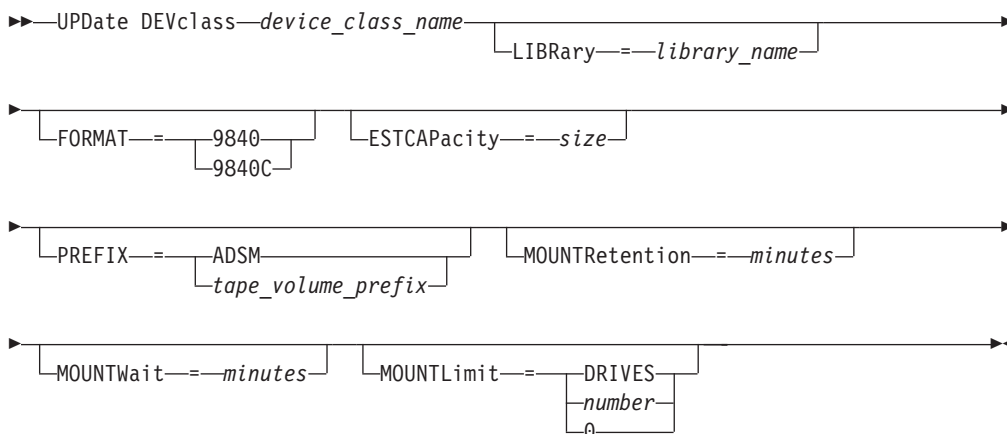
Specifies that no new transactions can gain access to the storage pool. Any current transaction will continue and complete, but new transactions will be terminated.

**UPDATE DEVCLASS (Update a VOLSAFE device class)**

Use the VOLSAFE device type to work with StorageTek VolSafe brand media and drives. This technology uses media that cannot be overwritten. Therefore, do not use these media for short-term backups of client files, the server database, or export tapes.

**Privilege class**

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

**Syntax****Parameters****device\_class\_name (Required)**

Specifies the name of the device class to be updated. The maximum length of the device class name is 30 characters.

**LIBRARY**

Specifies the name of the defined library object that contains the VolSafe drives that can be used by this device class. If any drives in a library are VolSafe-enabled, all drives in the library must be VolSafe-enabled. See “DEFINE DEVCLASS (Define a VOLSAFE device class)” on page 210 for additional information on the VolSafe device type.

**FORMAT**

Specifies the recording format to be used when writing data to sequential access media. This parameter is optional; the default value is DRIVE.

**Attention:** If you specify DRIVE for a device class that has non-compatible sequential access devices, then you must mount volumes on devices that are capable of reading or writing the format established when the volume was first mounted. This can cause delays if the only sequential access device that can access the volume is already in use.

The following table lists the recording formats and estimated capacities for VolSafe devices:



Table 364. Recording formats and default estimated capacities for volsafe tapes

| Format | Estimated Capacity | Description  |
|--------|--------------------|--|
| DRIVE  | –                  | The server selects the highest format that is supported by the drive on which a volume is mounted.<br><br><b>Attention:</b> Avoid specifying DRIVE when a mixture of drives is used within the same library. For example, do not use this option for a library containing some drives that support recording formats superior to other drives. |
| 9840   | 20 GB              | Uncompressed (standard) format, using a 20 GB cartridge with 270 meters (885 feet) of tape   |
| 9840C  | 80 GB              | LZ-1 Enhanced (4:1) compressed format, using a 80 GB cartridge with 270 meters (885 feet) of tape  |

**ESTCAPacity**

Specifies the estimated capacity for the volumes assigned to 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 **K** (kilobytes), **M** (megabytes), or **G** (gigabytes). The smallest value allowed is 100 KB (that is, ESTCAPACITY=100K).

For example, ESTCAPACITY=5M specifies that the estimated capacity for a volume in this device class is 5 MB.

For more information on the default estimated capacity for cartridge tapes, see Table 364.

**PREFIX**

Specifies the high-level qualifier of the data set name that the server writes into the sequential access media labels. For each sequential access volume assigned to this device class, the server uses this prefix to create the data set name. This parameter is optional.

If you have already established a tape label naming convention that supports your current tape management system, consider using a tape volume prefix that conforms to your naming conventions.

**MOUNTRetention**

Specifies the number of minutes to retain an idle sequential access volume before dismounting it. This parameter is optional. The default value is 60. You can specify a number from 0 to 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online. However, for EXTERNAL library types (that is, a library managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**MOUNTWait**

Specifies the maximum number of minutes the server will wait for an operator to respond to a request to either mount a volume in a drive in a manual library or check in a volume to be mounted in an automated library. 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 a number from 0 to 9999.

### **MOUNTLimit**

Specifies the maximum number of sequential access volumes that can be simultaneously mounted for the device class. This parameter is optional. The default is DRIVES. You can specify a number from 1 to 4096.

If you plan to use the simultaneous write function, ensure that sufficient drives are available for the write operation. If the number of drives needed for a simultaneous write operation is greater than the value of the MOUNTLIMIT parameter for a device class, the transaction will fail.

For details about the simultaneous write function, refer to the *Administrator's Guide*.

The following are possible values:

### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives defined to the library is used to calculate the true value (including online status).

**Note:** For EXTERNAL library types, do not use the DRIVES default for the MOUNTLIMIT value. Specify the number of drives for the library as the MOUNTLIMIT value.

### *number*

Specifies the maximum number of drives used concurrently in this device class by the server. This value must never be allowed to exceed the number of drives that are defined and online in the library that services this device class.

### **0 (zero)**

Specifies that no new transactions can gain access to the storage pool.

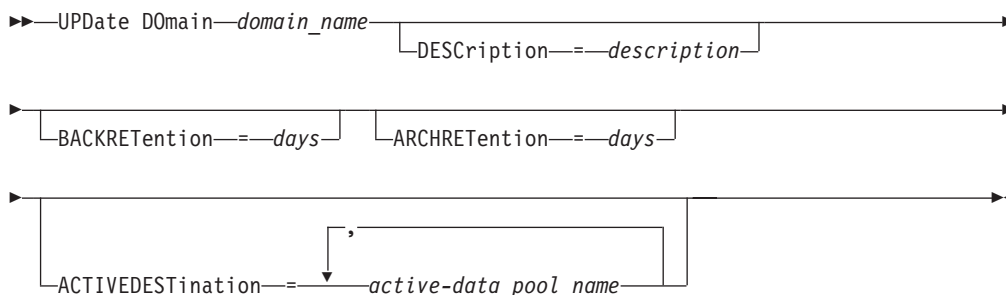
## UPDATE DOMAIN (Update a policy domain)

Use this command to change a policy domain.

### 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

#### *domain\_name* (Required)

Specifies the name of the policy domain.

#### DESCRiption

Describes the policy domain using a text string. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string (`""`).

#### BACKRETention

Specifies the number of days (from the date the backup versions became inactive) to retain backup versions that are no longer on the client file system. This parameter is optional. You can specify an integer from 0 to 9999. The server uses the backup retention value to manage inactive versions of files when any of the following conditions occur:

- A file is rebound to a new management class, 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

Specifies the number of days (from the date of archive) to retain archive copies. This parameter is optional. You can specify an integer from 0 to 30000. The server uses the archive retention value to manage archive copies of files when either of the following conditions occur:

- The management class to which a file is bound no longer exists, and the default management class does not contain an archive copy group.

## UPDATE DOMAIN

- 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.

### ACTIVEDESTINATION

Specifies the names of active-data pools that store active versions of backup data for nodes assigned to the domain. This parameter is optional. Spaces between the names of the active-data pools are not permitted. You cannot specify more than ten active-data pools for a domain.

Before the Tivoli Storage Manager server writes data to an active-data pool, it verifies that the node owning the data is assigned to a domain that has the active-data pool listed in the ACTIVEDESTINATION list. If the server verifies that the node meets this criteria, the data is stored in the active-data pool. If the node does not meet the criteria, then the data is not stored in the active-data pool. If the backup is performed with simultaneous write to an active-data pool, the server performs the verification during backup operations by Tivoli Storage Manager backup-archive clients or by application clients using the Tivoli Storage Manager API. The verification is also performed when active-data is being copied using the COPY ACTIVEDATA command.

### Example: Update the backup retention period for a policy domain

For the EMPLOYEE\_RECORDS policy domain, set the backup retention to 45 days and the archive retention to 75 days.

```
update domain employee_records  
backretention=45 archretention=75
```

### Related commands

*Table 365. Commands related to UPDATE DOMAIN*

| Command          | Description   |
|------------------|---|
| COPY DOMAIN      | Creates a copy of a policy domain.  |
| DEFINE DOMAIN    | Defines a policy domain that clients can be assigned to.                    |
| DEFINE POLICYSET | Defines a policy set within the specified policy domain.                    |
| DELETE DOMAIN    | Deletes a policy domain along with any policy objects in the policy domain. |
| QUERY DOMAIN     | Displays information about policy domains.                                  |

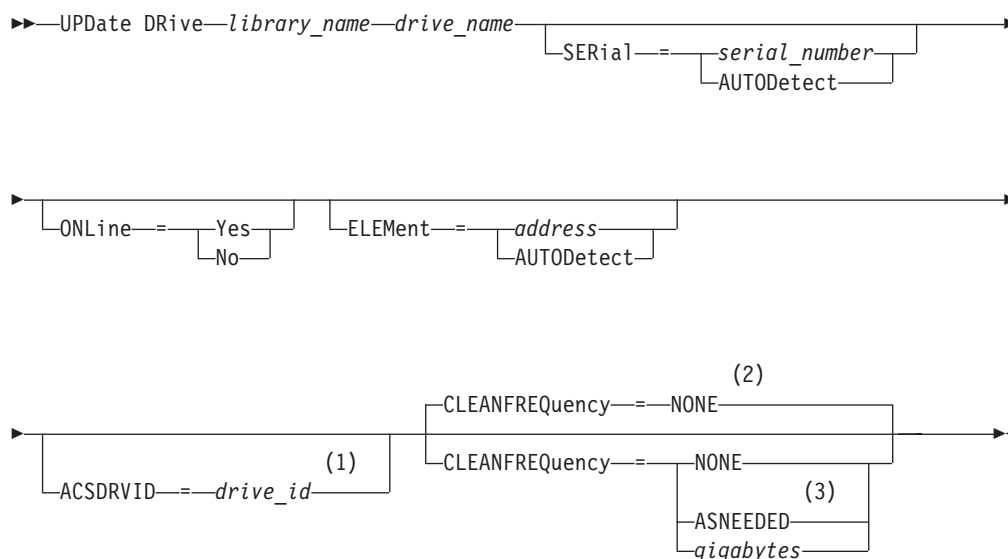
## UPDATE DRIVE (Update a drive)

Use this command to update a drive.

### Privilege class

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

### Syntax



### Notes:

- 1 The ACSDRVID parameter is valid only for drives in ACSLS libraries.
- 2 The CLEANFREQUENCY parameter is valid only for drives in SCSI libraries.
- 3 The CLEANFREQUENCY=ASNEEDED parameter value does not work for all tape drives. See the parameter description for more information.

### Parameters

#### *library\_name* (Required)

Specifies the name of the library to which the drive is assigned.

#### *drive\_name* (Required)

Specifies the name that is assigned to the drive.

#### SERIAL

Specifies the serial number for the drives being updated. This parameter is valid only for drives in SCSI libraries. This parameter is optional. The possible values are:

#### *serial\_number*

Specifies the serial number for the drive being updated.

**Note:** If a path to this drive has already been defined, then the number you enter here will be compared to the number detected by Tivoli Storage Manager. If the numbers do not match, then this command will fail.

### AUTODETECT

Specifies that the serial number will be automatically detected and used by Tivoli Storage Manager if a path has already been defined to this drive.

If a path to this drive has not been defined, then the serial number will not be detected.

### ONLine

Specifies whether the drive is available for use. This parameter specifies if drives can be taken offline and used for another activity, such as maintenance. This parameter is optional.

You can issue this command when the drive is involved in an active process or session, but it is not recommended. If you issue a command to take the drive offline while it is in use, an error message is issued. The mounted volume will complete its current process. If this volume was part of a series of volumes for a given transaction, the drive will not be available to complete mounting the series. If no other drives are available, the process fails.

**Attention:** When a drive is in use, do not specify the **ELEMENT** parameter with the **ONLINE** parameter. The drive will not be updated, and the command will fail.

The drive state is not changed even if the server is halted and restarted. If a drive is offline when the server is restarted, a warning message is issued stating that the drive must be manually brought online. If all the drives in a library have been updated to be offline, processes that need a library mount point will fail, rather than queue up for a mount point.

### YES

Specifies that the drive is available for use (online).

### No

Specifies that the drive is not available for use (offline).

### ELEMent

Specifies the element address of the drive within a SCSI library. The server uses the element address to connect the physical location of the drive to the SCSI address of the drive. This parameter is valid for a drive in a SCSI library when the command is issued from a Tivoli Storage Manager library manager server. The possible values are:

#### *address*

Specifies the element address for the drive being updated.

To find the element address for your library configuration, consult the manufacturer's information.

Additional information on element addresses can be found at the IBM Tivoli Storage Manager Web site: <http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html>. Follow the links to device information. Select the appropriate product and platform, then scroll to the required device. Click on a specific device to see element addresses.

**Remember:** If a path to this drive has already been defined, then the number you enter here will be compared to the number previously detected by Tivoli Storage Manager. If the numbers do not match, then this command will fail.

**AUTODETECT**

Specifies that the element number will be automatically detected and used by Tivoli Storage Manager if a path has already been defined to this drive.

If a path to this drive has not been defined, then the element number will not be detected.

**Restriction:** If the library in which the drive is located does not support the Read Element Status SCSI command, and ELEMENT=AUTODETECT, the command will fail with a Tivoli Storage Manager error message.

**ACSDRVID**

Specifies the ID of the drive being accessed in an ACSLS library. The drive ID is a set of numbers that indicates the physical location of a drive within an ACSLS library. This drive ID must be specified as *a,l,p,d*, where *a* is the ACSID, *l* is the LSM (library storage module), *p* is the panel number, and *d* is the drive ID. The server needs the drive ID to connect the physical location of the drive to the drive's SCSI address. See your StorageTek documentation for details.

**CLEANFREQuency**

Specifies how often the server activates drive cleaning. This parameter is optional. For the most complete automation of cleaning for an automated library, you must have a cleaner cartridge checked into the library's volume inventory. This parameter is valid only for drives in SCSI libraries, and not valid for externally managed libraries, such as 3494 libraries or StorageTek libraries managed under ACSLS.

For details on using this parameter for automated and manual libraries, see the *Administrator's Guide*.

**Important:** There are special considerations if you plan to use server-activated drive cleaning with a SCSI library that provides automatic drive cleaning support in its device hardware. See the *Administrator's Guide* for details.

**NONE**

Specifies that the server does not track cleaning for this drive. Use this parameter for libraries that have their own automatic cleaning.

**ASNEEDED**

Specifies that the server loads the drive with a checked-in cleaner cartridge only when a drive reports to the device driver that it needs cleaning.

The CLEANFREQUENCY=ASNEEDED parameter value does not work for all tape drives. To determine whether a drive supports this function, see the following Web site:

[www.ibm.com/software/sysmgmt/products/support/IBM\\_TSM\\_Supported\\_Devices\\_for\\_AIXHPSUNWIN.html](http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html)

At the Web site, click the drive to view detailed information. If **ASNEEDED** is not supported, you can use the *gigabytes* value for automatic cleaning.

**Restriction:** Tivoli Storage Manager does not control the drives connected to the NAS file server. If a drive is attached only to a NAS file server (no connection to a storage agent or server), do not specify **ASNEEDED** for the cleaning frequency.

*gigabytes*

Specifies, in gigabytes, how much data is processed on the drive before the

## UPDATE DRIVE

server loads the drive with a cleaner cartridge. The server resets the gigabytes-processed counter each time it loads a cleaner cartridge in the drive.

Consult the drive manufacturer's information for cleaning recommendations. If the information gives recommendations for cleaning frequency in terms of hours of use, convert to a gigabytes value by doing the following:

1. Use the bytes-per-second rating for the drive to determine a gigabytes-per-hour value.
2. Multiply the gigabytes-per-hour value by the recommended hours of use between cleanings.
3. Use the result as the cleaning frequency value.

**Tip:** For IBM 3590 and IBM 3570, specify a value for the cleaning frequency to ensure that the drives receive adequate cleaning. Consult the drive manufacturer's information for cleaning recommendations. Using the cleaning frequency recommended by IBM will not overclean the drives.

### Example: Update a drive's element address

Update DRIVE3, located in the library named AUTO, by changing the element address to 119.

```
update drive auto drive3 element=119
```

### Example: Take a drive offline

Update DRIVE3, located in the library MANLIB, to take it offline.

```
update drive manlib drive3 online=no
```

## Related commands

Table 366. Commands related to UPDATE DRIVE

| Command       | Description                                       |
|---------------|---|
| CLEAN DRIVE   | Marks a drive for cleaning.                       |
| DEFINE DRIVE  | Assigns a drive to a library.                     |
| DEFINE PATH   | Defines a path from a source to a destination.    |
| DELETE DRIVE  | Deletes a drive from a library.                   |
| QUERY DRIVE   | Displays information about drives.                |
| QUERY LIBRARY | Displays information about one or more libraries. |
| UPDATE PATH   | Changes the attributes associated with a path.    |



## UPDATE LIBRARY (Update a library)

Use this command to update a library definition.

To update the device name, the ACS number, or the external manager path name of a library, you must use the UPDATE PATH command.

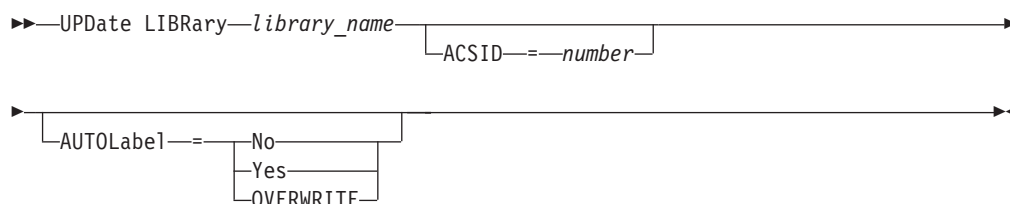
This section includes syntax diagrams for a number of different library configurations:

| Configuration task  | Syntax diagram   |
|---|--|
| LAN - Update an ACSLS library <sup>1</sup>                                    | "Syntax for an ACSLS library on a LAN"   |
| SAN - Update an ACSLS library <sup>1</sup>                                    | "Syntax for an ACSLS library to a library manager, on a SAN"   |
| SAN - Update a library (SCSI, 349X, FILE) defined to a library manager server | "Syntax for library sharing in a SAN, not for NDMP operations (update a library defined to a library manager)" on page 1060 (SAN - Update a library defined to a library manager)              |
| SAN - Update a library (SHARED) defined to a library client server            | "Syntax for library sharing in a SAN, not for NDMP operations (update a library defined to a library client)" on page 1060 (SAN - Update a library defined to a library client)                |
| SAN - Update a library used for NDMP operations                               | "Syntax for a library used for NDMP operations (update a library controlled directly by Tivoli Storage Manager)" on page 1061 (Update a library controlled directly by Tivoli Storage Manager) |
|   |  |

### Privilege class

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

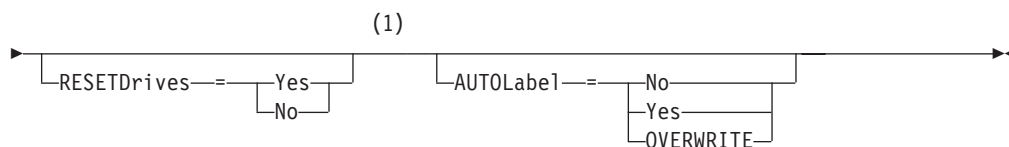
### Syntax for an ACSLS library on a LAN



### Syntax for an ACSLS library to a library manager, on a SAN



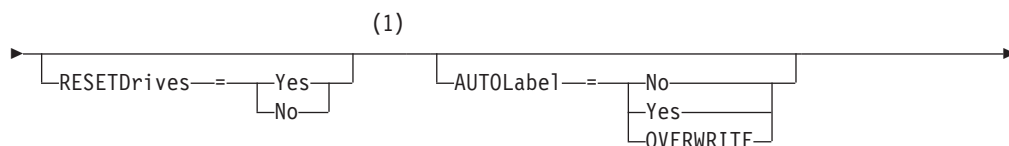
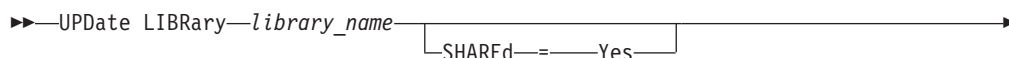
## UPDATE LIBRARY



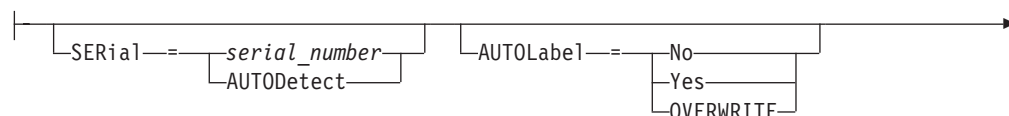
### Notes:

- 1 RESETDRIVES defaults to Yes only when a library is defined with SHARED=Yes; otherwise, the parameter defaults to No.

### Syntax for library sharing in a SAN, not for NDMP operations (update a library defined to a library manager)



### A (SCSI):



### B (349X):



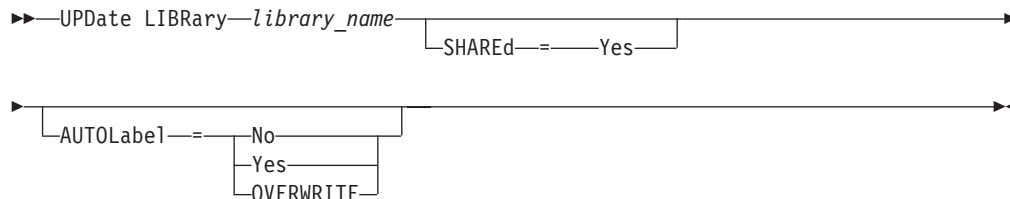
### Notes:

- 1 RESETDRIVES defaults to YES only when a library is defined with SHARED=YES; otherwise, the parameter defaults to No.

### Syntax for library sharing in a SAN, not for NDMP operations (update a library defined to a library client)



## Syntax for a library used for NDMP operations (update a library controlled directly by Tivoli Storage Manager)



### Parameters

#### *library\_name* (Required)

Specifies the name of the library to be updated.

#### SHARED

Specifies that this library is shared with other servers in a storage area network (SAN). You must issue this command from the server defined as the primary library manager for the shared library. This parameter is required for libraries defined to a library manager and for libraries used for NDMP operations. Specify SHARED=YES to update a library that is not currently shared.

**Important:** If a library has a path from a data mover (such as a NAS file server) but no connection to the Tivoli Storage Manager server, the library cannot be shared with another Tivoli Storage Manager server.

#### WORMSCRatchcategory

Indicates the category number used for WORM scratch volumes in a library. This applies to IBM 3494 or 3495 Tape Library Dataservers only.

**Attention:** You can specify a value for the **WORMSCRATCHCATEGORY** parameter only if you did not previously specify a value using the **DEFINE LIBRARY** command. To determine whether a value has been specified, issue the **QUERY LIBRARY** command. If the output field "WORM Scratch Category" does not contain a value, you can issue the **UPDATE LIBRARY** command to specify a value. However, if the field does contain a value, you cannot use the **UPDATE LIBRARY** command to change it.

#### SERial

Specifies the serial number for the library being updated. This parameter is valid only for SCSI libraries. This parameter is optional. The possible values are:

##### *serial\_number*

Specifies the serial number for the library being updated.

If a path to this library has already been defined, then the number you enter here will be compared to the number detected by Tivoli Storage Manager. If the numbers do not match, then this command will fail. If a path has not been defined, this serial number will be verified when a path is defined.

#### AUTODetect

Specifies that the serial number will be automatically detected and used by Tivoli Storage Manager if a path has already been defined to this library.

If a path to this library has not been defined, then the serial number will not be detected.

### **AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default for 349X, ACSLS, EXTERNAL, and MANUAL libraries is Yes. The default for SCSI libraries is NO.

To use this option, you need to check in the tapes with CHECKLABEL=BARCODE on the CHECKIN LIBVOLUME command.

#### **No**

Specifies that the server does not attempt to label any volumes.

#### **Yes**

Specifies that the server only labels unlabeled volumes.

### **OVERWRITE**

Specifies that the server attempts to overwrite an existing label. The server overwrites existing labels *only* if both the existing label and the bar code label are not already defined in any server storage pool or volume history list.

### **RELABELSCRatch**

Specifies whether the server relabels volumes that have been deleted and returned to scratch. When this parameter is set to Yes, a LABEL LIBVOLUME operation is started and the existing volume label is overwritten. This parameter is optional and intended for use with a Virtual Tape Library (VTL).

**Note:** If you have both virtual and real volumes in your VTL, both types will be relabeled when this parameter is enabled. If the VTL includes real volumes, specifying this option can impact performance.

#### **No**

Specifies that the server does not relabel volumes that are deleted and returned to scratch.

#### **Yes**

Specifies that the server relabels volumes that are deleted and returned to scratch.

### **PRIMarylibmanager**

Specifies the name of the IBM Tivoli Storage Manager server that is responsible for controlling access to library resources. You must define this server with the DEFINE SERVER command before you can use it for a primary library manager. This parameter is only valid for libraries defined to a library client.

### **ACSID**

Specifies the number of this StorageTek library assigned by the ACSSA (Automatic Cartridge System System Administrator). This can be a number from 0 to 126. Issue QUERY ACS on your system to get the number for your library ID. See your StorageTek documentation for more information.

### **RESETDrives**

Specifies whether the server performs a target reset when the server is restarted or when a library client or storage agent re-connection is established.

**Attention:** This parameter only applies to SCSI, 3494, MANUAL, and ACSLS type libraries.

#### **Yes**

Specifies that the target reset is to be performed. YES is the default for SCSI, 3494, MANUAL, and ACSLS libraries defined or updated with SHARED=YES.

**No**

Specifies that the target reset is not performed. NO is the default for SCSI, 3494, MANUAL, and ACSLS libraries defined with SHARED=NO.

**Example: Add new devices to a shared library**

Update a 3494 shared library named 3494LIB2 with new device names.

```
update library 3494lib2 device=/dev/lmcp1,/dev/lmcp2,/dev/lmcp3
```

**Example: Update an ID number for an ACSLS library**

Update an ACSLS library named ACSLSLIB with a new ID number.

```
update library acslslib acsid=1
```

**Example: Update an external library's path name**

Update an external library named EXTLIB with a new path name for the media manager.

```
update library extlib externalmanager=/v/server/mediamanager
```

**Example: Change the library manager server for a library**

For a library client server, change the name of the library manager server to CASTOR.

```
update library ltolib primarylibmanager=castor
```

**Related commands**

*Table 367. Commands related to UPDATE LIBRARY*

| Command        | Description   |
|----------------|---|
| AUDIT LIBRARY  | Ensures that an automated library is in a consistent state.         |
| DEFINE DRIVE   | Assigns a drive to a library.                                       |
| DEFINE LIBRARY | Defines an automated or manual library.                             |
| DEFINE PATH    | Defines a path from a source to a destination.                      |
| DELETE DRIVE   | Deletes a drive from a library.                                     |
| DELETE LIBRARY | Deletes a library.  |
| DELETE PATH    | Deletes a path from a source to a destination.                      |
| QUERY DRIVE    | Displays information about drives.                                  |
| QUERY LIBRARY  | Displays information about one or more libraries.                   |
| QUERY PATH     | Displays information about the path from a source to a destination. |
| UPDATE DRIVE   | Changes the attributes of a drive.                                  |
| UPDATE PATH    | Changes the attributes associated with a path.                      |

## UPDATE LIBVOLUME (Change the status of a storage volume)

Use this command to change the status of a sequential access storage volume in a library.

### Privilege class

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

### Syntax

```

►►—UPDate LIBVolume—library_name—volume_name—┐
                                                    └─STATUS—=—┐
                                                            └─PRIVate—┐
                                                                └─SCRatch—┐
                                                                    └─►
└─OWNER—=—server_name—┐
                        └─►
  
```

### Parameters

#### *library\_name* (Required)

Specifies the name of the library.

#### *volume\_name* (Required)

Specifies the volume name of the storage volume.

#### STATus

Specifies a change to the status of a storage volume. Possible values are:

##### PRIVate

Specifies that the server updates the storage volume to a private volume.

##### SCRatch

Specifies that the server updates the storage volume to a scratch volume.

#### OWNer

Specifies which server owns a private volume in a shared library that is shared across a SAN. You can change the owner of a private volume in a shared library (SAN) when you issue the command from the library manager server. If you do not specify this parameter, the library manager server owns the private volume.

**Important:** Do not use OWNER as a value for scratch volumes. However, you can use OWNER when changing a scratch volume to private.

### Example: Update a volume's status

Update the volume named WPDV00 located in the library named AUTO to reflect a status of PRIVATE.

```
update libvolume auto wpdv00 status=private
```

### Related commands

Table 368. Commands related to UPDATE LIBVOLUME

| Command       | Description   |
|---------------|---|
| AUDIT LIBRARY | Ensures that an automated library is in a consistent state. |

*Table 368. Commands related to UPDATE LIBVOLUME (continued)*

| Command            | Description  |
|--------------------|--|
| CHECKIN LIBVOLUME  | Checks a storage volume into an automated library.                       |
| CHECKOUT LIBVOLUME | Checks a storage volume out of an automated library.                     |
| DEFINE VOLUME      | Assigns a volume to be used for storage within a specified storage pool. |
| LABEL LIBVOLUME    | Labels volumes in manual or automated libraries.                         |
| QUERY LIBRARY      | Displays information about one or more libraries.                        |
| QUERY LIBVOLUME    | Displays information about a library volume.                             |

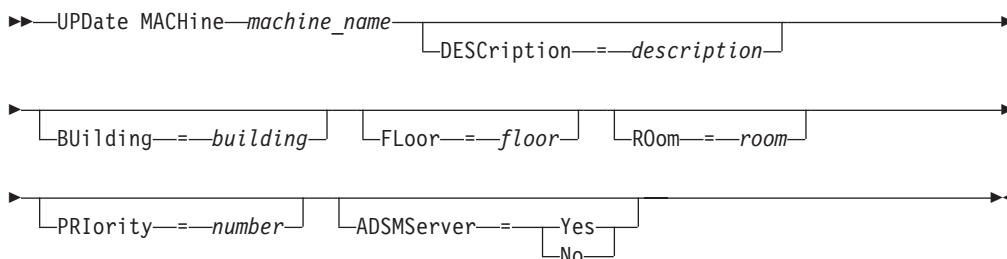
## UPDATE MACHINE (Update machine information)

Use this command to update machine information. This information will be included in the plan file to help you to recover the client machines.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *machine\_name* (Required)

Specifies the name of the machine to be updated.

#### DESCRiption

Specifies a description of the machine. This parameter is optional. The text can be up to 255 characters. Enclose the text in quotation marks if it contains any blank characters. To remove existing text, specify a null string (`""`).

#### BUiLding

Specifies the name or number of the building that this machine is in. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters. To remove existing text, specify a null string (`""`).

#### FLoor

Specifies the name or number of the floor that this machine is on. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters. To remove existing text, specify a null string (`""`).

#### ROom

Specifies the name or number of the room that this machine is in. This parameter is optional. The text can be up to 16 characters. Enclose the text in quotation marks if it contains any blank characters. To remove existing text, specify a null string (`""`).

#### PRIority

Specifies the restore priority for the machine as an integer from 1 to 99. The highest priority is 1. This parameter is optional. Tivoli Storage Manager uses this value to prioritize client machine recovery.

#### ADSMSeRver

Specifies whether the machine contains a Tivoli Storage Manager server. This parameter is optional. Possible values are:

##### No

This machine does not contain a Tivoli Storage Manager server.



Yes

This machine contains a Tivoli Storage Manager server. Only one machine can be defined as containing a Tivoli Storage Manager server.

### Example: Update information for a specific machine

Update the DISTRICT5 machine information to reflect that it contains the server.

```
update machine district5 adsmserver=yes
```

### Related commands

Table 369. Commands related to UPDATE MACHINE

| Command        | Description  |
|----------------|--|
| DEFINE MACHINE | Defines a machine for DRM.   |
| DELETE MACHINE | Deletes a machine.   |
| INSERT MACHINE | Inserts machine characteristics or recovery instructions into the IBM Tivoli Storage Manager database. |
| QUERY MACHINE  | Displays information about machines.   |

## UPDATE MGMTCLASS (Update a management class)

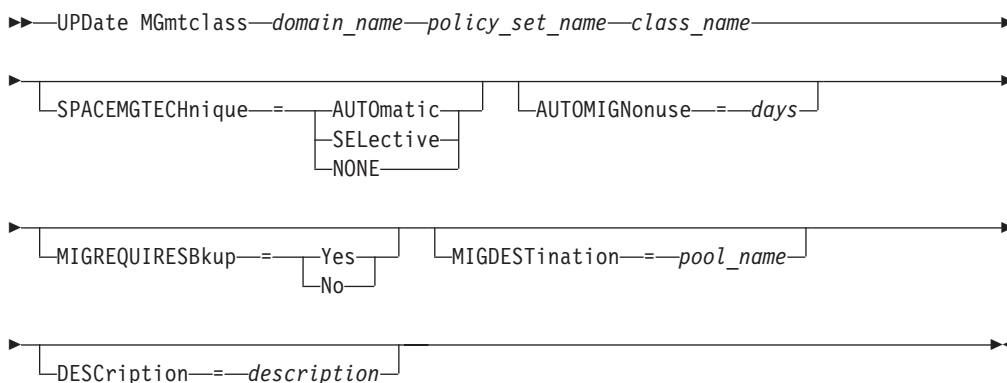
Use this command to change a management class. To allow clients to use the updated management class, you must activate the policy set that contains the management class.

**Important:** The UPDATE MGMTCLASS command fails if a copy storage pool is specified as the destination for files that were migrated by a Tivoli Storage Manager for Space Management client.

### 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

#### *domain\_name* (Required)

Specifies the policy domain to which the management class belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set to which the management class belongs. You cannot update a management class that belongs to the ACTIVE policy set.

#### *class\_name* (Required)

Specifies the management class to update.

#### SPACEMGTECHnique

Specifies whether a file using this management class is eligible for migration. This parameter is optional. This parameter is effective only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients. Possible values are:

##### **AUTOMatic**

Specifies that the file is eligible for both automatic migration and selective migration.

##### **SElective**

Specifies that the file is eligible for selective migration only.

##### **NONE**

Specifies that the file is not eligible for migration.

**AUTOMIGNonuse**

Specifies the number of days that must elapse since a file was last used before it is eligible for automatic migration. This parameter is optional. If **SPACEMGTECHNIQUE** is not **AUTOMATIC**, the server ignores this attribute. You can specify an integer from 0 to 9999.

This parameter is effective only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients.

**MIGREQUIRESBkup**

Specifies whether a backup version of a file must exist before a file can be migrated. This parameter is optional. This parameter is effective only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients. Possible values are:

**Yes**

Specifies that a backup version must exist.

**No**

Specifies that a backup version is optional.

**MIGDESTination**

Specifies the primary storage pool where the server initially stores files migrated by Tivoli Storage Manager for Space Management clients. This parameter is effective only for Tivoli Storage Manager for Space Management clients, not for backup-archive clients or application clients.

The command fails if you specify a copy storage pool as the destination.

**DESCription**

Specifies a description of the management class. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("").

**Example: Update the policy domain and storage pool of a specific management class**

For the management class **ACTIVEFILES**, in policy set **VACATION** in the **EMPLOYEE\_RECORDS** policy domain, change the storage pool where migrated files are stored.

```
update mgmtclass employee_records vacation
activefiles migdestination=diskpool2
```

**Related commands**

*Table 370. Commands related to UPDATE MGMTCLASS*

| Command             | Description  |
|---------------------|--|
| ASSIGN DEFMGMTCLASS | Assigns a management class as the default for a specified policy set.                      |
| COPY MGMTCLASS      | Creates a copy of a management class.  |
| DEFINE COPYGROUP    | Defines a copy group for backup or archive processing within a specified management class. |
| DEFINE MGMTCLASS    | Defines a management class.  |
| DEFINE POLICYSET    | Defines a policy set within the specified policy domain.                                   |

## UPDATE MGMTCLASS

*Table 370. Commands related to UPDATE MGMTCLASS (continued)*

| Command          | Description   |
|------------------|---|
| DELETE MGMTCLASS | Deletes a management class and its copy groups from a policy domain and policy set. |
| QUERY COPYGROUP  | Displays the attributes of a copy group.  |
| QUERY MGMTCLASS  | Displays information about management classes.                                      |
| QUERY POLICYSET  | Displays information about policy sets.   |
| UPDATE COPYGROUP | Changes one or more attributes of a copy group.                                     |

## UPDATE NODE (Update node attributes)

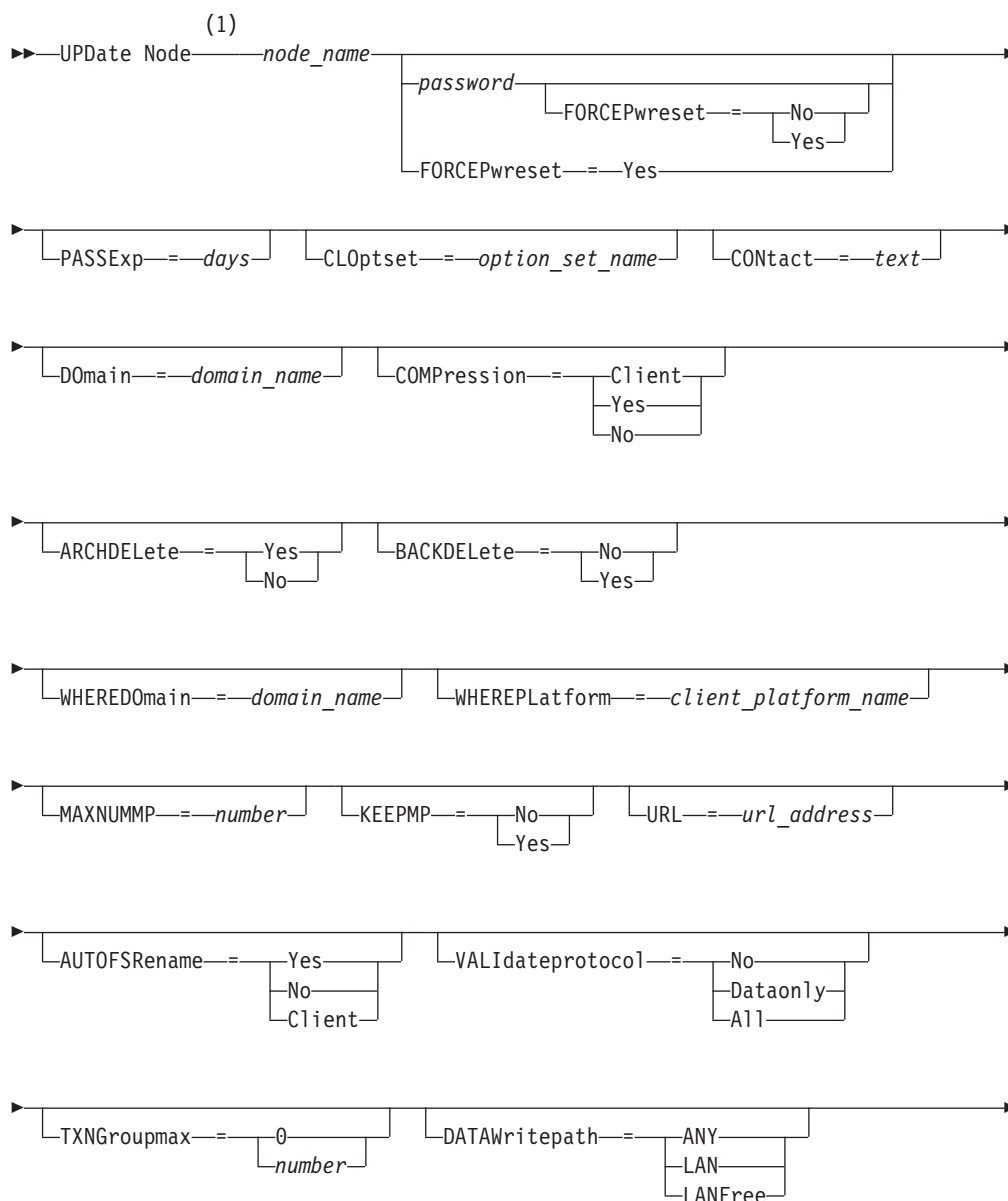
Use this command to modify the attributes of a registered node.

You must use the RENAME NODE command to change the name of a registered node.

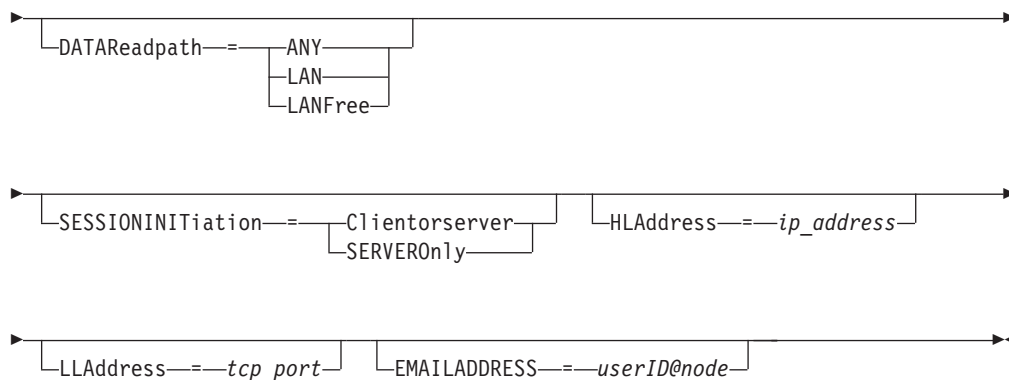
### Privilege class

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



## UPDATE NODE



### Notes:

- 1 You must specify at least one optional parameter on this command.

### Parameters

#### *node\_name* (Required)

Specifies the name of the client node to be updated. You can use wildcard characters to specify this name.

#### *password*

Specifies the new password for the client node. This parameter is optional. The maximum length of the password is 64 characters. Passwords remain current for a period determined by the password expiration period.

#### **FORCEPwreset**

Specifies whether to force a client to change or reset the password. This parameter is optional. Possible values are:

##### **No**

Specifies that the password expiration period is set by the SET PASSEXP command. The administrator does not force a client to change or reset the password while attempting to logon to the server.

**Remember:** This value is valid only when you specify a password.

##### **Yes**

Specifies that the client node or administrator's password will expire at the next logon. The client must change or reset the password at that time.

#### **PASSExp**

Specifies the number of days the password remains valid. You can set the password expiration period from 0 to 9999 days. A value of 0 means that the password never expires. This parameter is optional. If you do not specify this parameter, the password expiration period is unchanged.

You can change the password expiration period by using the UPDATE NODE or SET PASSEXP commands. The SET PASSEXP command enables you to set a common expiration period for all administrators and client nodes, or you may use it to selectively set password expiration periods. If you selectively set a password expiration period by using the REGISTER NODE command, the UPDATE NODE command, or the SET PASSEXP command, the expiration period is excluded from common password expiration periods that were created using the SET PASSEXP command.

You can use the RESET PASSEXP command to reset the password expiration period to the common expiration period.

**CLOptset**

Specifies the name of the option set to be used by the client. This parameter is optional.

**CONtact**

Specifies a text string of information identifying the client node. This parameter is optional. The maximum length of the text string is 255 characters. Enclose the contact information in quotation marks if it contains any blanks. To remove previously defined contact information, specify a null string ("").

**DOmain**

Specifies the name of the policy domain to which you want to register the client node. This parameter is optional.

**Restriction:** For servers with data retention protection enabled, an archived registered node cannot be reassigned to a different policy domain. See the *Administrator's Guide* for more information.

**COMPression**

Specifies whether the client node compresses its files before sending them to the server for backup and archive. This parameter is optional.

**Restriction:** This parameter cannot be specified for a NAS node.

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.

**ARCHDElete**

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.

**BACKDElete**

Specifies whether the client node can delete its own backup files from the server. This parameter is optional. 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.

### **WHEREDomain**

Specifies the name of the policy domain to be used as a filter in combination with the node name to select nodes to update. This parameter is optional.

### **WHEREPlatform**

Specifies the name of the client platform to be used as a filter in combination with the node name to select nodes to update. This parameter is optional.

### **MAXNUMMP**

Specifies the maximum number of mount points a node can use on the server or storage agent only for operations such as backup, archive, and Tivoli Storage Manager for Space Management migration. The parameter is optional and does not apply to nodes with a type of NAS or SERVER. The default value is 1. You can specify an integer from 0–999. A value of 0 specifies that a node cannot acquire any mount point for a client data store operation. The **MAXNUMMP** value is not evaluated or enforced during client data read operations such as restore, retrieve, and Tivoli Storage Manager for Space Management recall. However, mount points in use for data read operations are evaluated against attempted concurrent data store operations for the same client node and may prevent the data store operations from being able to acquire mount points.

For volumes in a storage pool associated with the FILE device type, the server can have multiple sessions to read and one process to write to the same volume concurrently. To provide efficient access for nodes with data in FILE-type storage pools, increase the value of the **MAXNUMMP** parameter to for greater concurrency when accessing FILE volumes.

For nodes that store data into primary storage pools with simultaneous write enabled, you may need to adjust the value of the **MAXNUMMP** parameter to specify the correct number of mount points for each client session. For details, refer to information about simultaneous write in the *Administrator's Guide*.

### **URL**

Specifies the client's Uniform Resource Locator (URL) address that the administrator can use to administer the Tivoli Storage Manager client. If you want to remove the value for this field, specify empty single quotation marks or empty double quotation marks with no spaces (" for single quotation marks, or "" for double quotation marks).

### **KEEPMP**

Specifies whether the client node keeps the mount point for the entire session. The parameter is optional. Possible values are:

#### **No**

Specifies that the client node release the mount point during the session. If policy definitions cause data to be stored to a disk storage pool after storing data to a sequential access storage pool, any mount points held by the session will be released.

#### **Yes**

Specifies that the client node must retain the mount point during the entire session. If policy definitions cause data to be stored to a disk storage pool after storing data to a sequential access storage pool, any mount points held by the session will not be released.

### **AUTOFSRename**

Specifies whether the client is prompted for renaming file spaces when the client system upgrades to a client that supports Unicode. The prompting and renaming, if allowed, occur only when the client runs one of the following



operations: archive, selective backup, full incremental backup, or partial incremental backup. The renaming changes the names of existing backed-up file spaces that are not in Unicode in server storage. Then the file spaces are backed up in Unicode. You can use this parameter for Unicode-enabled Tivoli Storage Manager clients using Windows, Macintosh OS X, and NetWare operating systems.

**Important:** After the client with support for Unicode is installed, any new file spaces that the client backs up are stored in server storage using the UTF-8 code page. UTF-8 is a byte-oriented encoding form specified by the Unicode Standard.

Possible values are:

#### Yes

The server automatically renames existing file spaces when the client system upgrades to a client that supports Unicode, and the client runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup. The renaming occurs whether the client uses the graphical user interface, the command line, or the client scheduler.

For example, the server renames a drive as follows:

|                |             |
|----------------|-------------|
| Original name: | D_DRIVE     |
| New name:      | D_DRIVE_OLD |

The new name indicates that the file space is stored on the server in format that is not Unicode.

#### No

The server does not rename file spaces automatically when the client system upgrades to a client that supports Unicode, and the client runs one of the following operations: archive, selective backup, full incremental backup, or partial incremental backup.

#### Client

The option AUTOFSRENAME in the client's option file determines whether file spaces are renamed.

By default, the client option is set to PROMPT. When the client system upgrades to a client that supports Unicode and the client runs a Tivoli Storage Manager operation with the graphical user interface or the command line, the program displays a one-time prompt to the user about whether to rename file spaces.

When the client scheduler runs an operation, the program does not prompt for a choice about renaming, and does not rename file spaces. Backups of existing file spaces are sent as before (not in Unicode).

Refer to the appropriate *Backup-Archive Clients Installation and User's Guide* book for more information on the client option.

#### VALIDateprotocol

Specify whether Tivoli Storage Manager performs a cyclic redundancy check to validate the data sent between the client and the server. The parameter is optional. Possible values are:

#### No

Specifies that Tivoli Storage Manager does not perform data validation on any data sent between the client and server.

### **Dataonly**

Specifies that Tivoli Storage Manager performs data validation only on file data that is sent between the client and server. This does not include the file metadata. This mode impacts performance because additional overhead is required to calculate and compare cyclic redundancy check (CRC) values between the client and the server.

### **All**

Specifies that Tivoli Storage Manager performs data validation on all client file data, client file metadata, and server metadata that is sent between the client and server. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the client and the server.

### **TXNGroupmax**

Specifies the number of files that are transferred as a group between a client and a server between transaction commit points. Client performance may be improved by using a larger value for this option.

Specifying 0 indicates the node will use the server global value that is set in the server options file. To use a value other than the server global value, specify a value of 4 through 65,000 for this parameter. The node value takes precedence over the server value.

**Tip:** Increasing the **TXNGROUPMAX** value will result in increased recovery log utilization. Higher recovery log utilization may increase the risk of running out of log space. Evaluate each node's performance before changing this parameter. For more information on managing the recovery log, see the *Administrator's Guide*.

### **DATAWritepath**

Specifies the transfer path used when the client sends data to the server, storage agent, or both, during storage operations such as backup or archive. The parameter is optional.

**Remember:** If a path is unavailable, the node cannot send any data. For example, if you select the LAN-free option but a LAN-free path is not defined, the operation will not work.

Possible values are:

#### **ANY**

Specifies that data is sent to the server, storage agent, or both, using any available path. A LAN-free path will be used if one is available. If a LAN-free path is unavailable, the data will be moved using the LAN.

#### **LAN**

Specifies that data is sent using the LAN.

#### **LANFree**

Specifies that data is sent using a LAN-free path.

### **DATAReadpath**

Specifies the transfer path used when the server, storage agent, or both read data for a client, during operations such as restore or retrieve. The parameter is optional.

**Remember:** If a path is unavailable, data cannot be read. For example, if you select the LAN-free option but a LAN-free path is not defined, the operation will not work.

Possible values are:

**ANY**

Specifies that the server, storage agent, or both use any available path to read data. A LAN-free path will be used if one is available. If a LAN-free path is unavailable, the data will be read using the LAN.

**LAN**

Specifies that data is read using the LAN.

**LANFree**

Specifies that data is read using a LAN-free path.

**SESSIONINITiation**

Controls whether the server or the client initiates sessions. The parameter is optional.

**Clientorserver**

Specifies that the client may initiate sessions with the server by communicating on the TCP/IP port defined with the server option TCPPORT. Server-prompted scheduling may also be used to prompt the client to connect to the server.

**SERVEROnly**

Specifies that the server will not accept client requests for sessions. All sessions must be initiated by server-prompted scheduling on the port defined for the client with the REGISTER or UPDATE NODE commands. You cannot use the client acceptor (dsmcad) to start the scheduler when **SESSIONINITIATION** is set to **SERVERONLY**.

**HLAddress**

Specifies the client IP address that the server contacts to initiate scheduled events. This optional parameter is used only when **SESSIONINITIATION** is set to **SERVERONLY**, regardless of any addresses previously used by the client to contact the server. If **SESSIONINITIATION SERVERONLY** is not in use, this option has no effect.

The address can be specified either in numeric or host name format. If a numeric address is used, it will be saved without verification by a domain name server. If the address is not correct, it can cause failures when the server attempts to contact the client. Host name format addresses will be verified with a domain name server. Verified names will be saved and resolved with Domain Name Services when the server contacts the client.

**LLAddress**

Specifies the client port number on which the client listens for sessions from the server. This optional parameter is used only when **SESSIONINITIATION** is set to **SERVERONLY**, regardless of any addresses previously used by the client to contact the server. If **SESSIONINITIATION SERVERONLY** is not in use, this option has no effect.

The value for this parameter must match the value of client option **TCPCLIENTPORT**. The default value is 1501.

**EMAILADDRESS**

This parameter is used by IBM Tivoli Storage Manager Express only. It is not currently used by Tivoli Storage Manager. Tivoli Storage Manager Express uses this parameter to specify the e-mail address or addresses for this administrator. Any reports that are directed to this administrator are sent to these e-mail addresses in HTML format.

### Example: Update a node's backup to compress data and keep the client from deleting archived files

Update node LARRY so that the data on node LARRY will be compressed when it is backed up or archived by Tivoli Storage Manager and so that the client cannot delete archived files.

```
update node larry compression=yes archdelete=no
```

### Example: Update a node's number of files that can be transferred as a group

Update node LARRY and increase the TXNGroupmax to 1,000.

```
update node larry txngroupmax=1000
```

### Related commands

Table 371. Commands related to UPDATE NODE

| Command        | Description   |
|----------------|---|
| QUERY NODE     | Displays partial or complete information about one or more clients.                 |
| REGISTER ADMIN | Defines a new administrator without granting administrative authority.              |
| REGISTER NODE  | Defines a client to the server and sets options for that user.                      |
| REMOVE NODE    | Removes a client from the list of registered nodes for a specific policy domain.    |
| RENAME NODE    | Changes the name for a client node.   |
| RESET PASSEXP  | Resets the password expiration for nodes or administrators.                         |
| SET PASSEXP    | Specifies the number of days after which a password is expired and must be changed. |
| UPDATE ADMIN   | Changes the password or contact information associated with any administrator.      |

## UPDATE NODEGROUP (Update a node group)

Use this command to modify the description of a node group.

### Privilege class

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

### Syntax

```
►►—UPDate NODEGroup—group_name—DESCription—=description—◄◄
```

### Parameters

*group\_name*

Specifies the name of the node group whose description you want to update.

DESCription

Specifies a description of the node group. This parameter is required. The maximum length of the description is 255 characters. If the description contains any blanks, enclose the entire description in quotation marks.

### Example: Update a node group's description

Update the node group, group1, with a new description.

```
update nodegroup group1 description="Human Resources"
```

### Related commands

Table 372. Commands related to UPDATE NODEGROUP

| Command                | Description   |
|------------------------|---|
| DEFINE BACKUPSET       | Defines a previously generated backup set to a server.  |
| DEFINE NODEGROUP       | Defines a group of nodes.                               |
| DEFINE NODEGROUPMEMBER | Adds a client node to a node group.                     |
| DELETE BACKUPSET       | Deletes a backup set.                                   |
| DELETE NODEGROUP       | Deletes a node group.                                   |
| DELETE NODEGROUPMEMBER | Deletes a client node from a node group.                |
| GENERATE BACKUPSET     | Generates a backup set of a client's data.              |
| QUERY BACKUPSET        | Displays backup sets.                                   |
| QUERY NODEGROUP        | Displays information about node groups.                 |
| UPDATE BACKUPSET       | Updates a retention value associated with a backup set. |

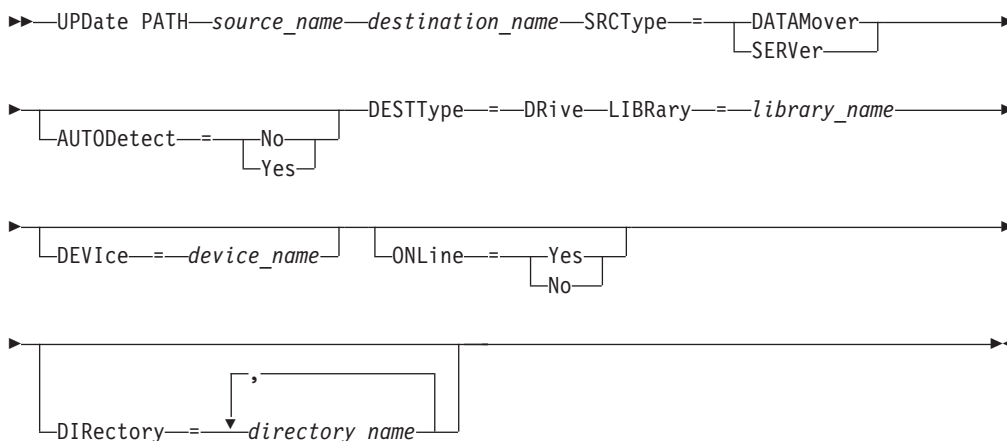
## UPDATE PATH (Change a path)

Use this command to update a path definition.

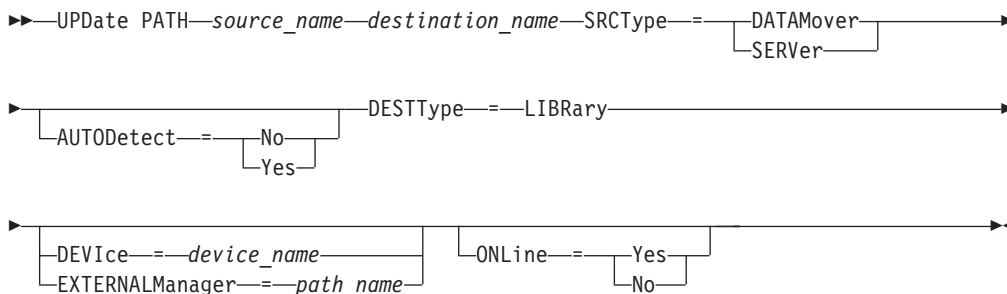
### Privilege class

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

### Syntax when the destination is a drive



### Syntax when the destination is a library



### Parameters

#### *source\_name* (Required)

Specifies the name of source for the path. This parameter is required.

#### *destination\_name* (Required)

Specifies the name of the destination. This parameter is required.

The destination can be a drive or a library.

**Important:** To define a path from a NAS data mover to a library, the library must have LIBTYPE of SCSI, 349x, or Automated Cartridge System Library Software (ACSLs).

#### SRCType (Required)

Specifies the type of the source. This parameter is required. Possible values are:

##### DATAMover

Specifies that a data mover is the source.

**SERVer**

Specifies that a server or a storage agent is the source.

**AUTODetect**

Specifies whether the serial number for a drive or library will be automatically detected, reported, and updated in IBM Tivoli Storage Manager. This parameter is optional. This parameter is only valid for paths defined from the local server to a drive or a library. Possible values are:

**No**

Specifies that the serial number will not be automatically updated.

**Yes**

Specifies that the serial number will be automatically updated to reflect the same serial number that the drive reports to IBM Tivoli Storage Manager.

**Important:**

1. If you have not previously entered a serial number, then AUTODETECT defaults to YES. If you have previously entered a serial number, then AUTODETECT defaults to NO.
2. AUTODETECT=YES in this command overrides the serial number set in the DEFINE DRIVE command.
3. **DESTTYPE=DRIVE only:** If you set DESTTYPE=DRIVE and AUTODETECT=YES, then the drive element number in the IBM Tivoli Storage Manager database will be automatically changed to reflect the same element number that corresponds to the serial number of that drive. This is true for drives in a SCSI library. For more information about the element number, see the DEFINE DRIVE command.
4. Depending on the capabilities of the device, the AUTODETECT parameter may not be supported.

**DESTType (Required)**

Specifies the type of the destination. This parameter is required. Possible values are:

**DRive**

Specifies that a drive is the destination. When the destination is a drive, you must specify a library name.

**LIBRARY**

Specifies that a library is the destination.

**LIBRARY**

Specifies the name of the library to which the drive is assigned. This parameter is required if DESTTYPE=DRIVE. The library and its drives must already be defined to the IBM Tivoli Storage Manager server. If the path is from a NAS data mover to a library, the library must have LIBTYPE of SCSI, 349x, or ACSLS.

**DEVIce**

Specifies the name of the device as known to the source, or FILE if the device is a logical drive in a FILE library.

The source uses the device name to access the drive or library. See Table 373 for examples.

*Table 373. Examples of device names*

| Source to destination                | Example   |
|--------------------------------------|-----------|
| Server to a drive (not a FILE drive) | /dev/rmt3 |

Table 373. Examples of device names (continued)

| Source to destination  | Example   |
|--|---|
| Server to a library  | /dev/lb4  |
| Storage agent to a drive (not a FILE drive)                                  | mt3   |
| Storage agent to a drive when the drive is a logical drive in a FILE library | FILE  |
| NAS data mover to a library  | mc0   |
| NAS data mover to a drive  | NetApp NAS file server: rst01<br><br>EMC Celerra NAS file server:<br>c436t011<br><br>IBM System Storage N Series: rst01 |

**Important:**

- For more complete information about device names when the source is a server, see the *Administrator's Guide*.
- For information about the device name when the source is a storage agent, see the *Storage Agent User's Guide*.
- For 349X libraries, the alias name is a symbolic name that is specified in the /etc/ibmatl.conf file. For more information, refer to the *IBM Tape Device Drivers Installation and User's Guide*. The Guides can be downloaded from the FTP site at ftp://ftp.software.ibm.com/storage/devdrv/. They are located in the Doc folder.
- For information about how to obtain names for devices that are connected to a NAS file server, consult the product information for the file server. For example, for a NetApp file server, connect to the file server using Telnet and issue the SYSCONFIG command. Use this command to determine device names for drives:

```
sysconfig -t
```

Use this command to determine the device name for a library:

```
sysconfig -m
```

**EXTERNALManager**

Specifies the location of the external library manager where IBM Tivoli Storage Manager can send media access requests. Use single quotation marks around the value of this parameter. For example, enter: 'c:\Program Files\bin\elm.exe'

This parameter is required when the library name is an external library.

**ONLine**

Specifies whether the path is available for use. This parameter is optional. Possible values are:

**Yes**

Specifies that the path is available for use.

**No**

Specifies that the path is not available for use.

The source and the destination must both be available to use the path.



For example, if the path from a data mover to a drive is online, but either the data mover or the drive is offline, you cannot use the path.

**Important:** If the path to a library is offline, the server will not be able to access the library. If the server is halted and restarted while the path to the library is offline, the library will not be initialized.

For additional information, see the *Administrator's Guide*.

### DIRectory

Specifies the directory location or locations for a storage agent to access the files in a FILE library. The DIRECTORY parameter is also used for devices of type REMOVABLEFILE. For REMOVABLEFILE devices, the DIRECTORY parameter provides information for the server (not a storage agent) along with the DRIVE parameter to describe access to the device. This parameter is optional.

On storage agents, this parameter is only valid when *all* of the following conditions are true:

- The source type is SERVER (meaning a storage agent that has been defined as a server to this server).
- The source name is the name of a storage agent, *not* the server.
- The destination is a logical drive that is part of a FILE library.
- If multiple directories were specified for the device class associated with the FILE library, the same number of directories must be specified with the DIRectory parameter of the DEFINE PATH command, for each drive in the FILE library. Storage agent directories are not validated on the Tivoli Storage Manager server. Specifying incorrect directories can cause a run-time failure.

The directory name or names identify the locations where the storage agent reads and writes the files that represent storage volumes for the FILE device class that is associated with the FILE library. The default value for DIRECTORY is the directory of the server at the time the command is issued.

Use a naming convention that you can use to associate the directory with a particular physical drive. This can help ensure that your configuration is valid for sharing the FILE library between the server and storage agent. If the storage agent is on a Windows system, use a universal naming convention (UNC) name. When the storage agent lacks permission to access remote storage, the storage agent will experience mount failures.

### Important:

- Tivoli Storage Manager does not create shares or permissions, or mount the target file system. You must perform these actions before starting the storage agent.
- You can modify a list of directories only by replacing the entire list.
- You must ensure that storage agents can access newly created FILE volumes. To access FILE volumes, storage agents replace names from the directory list in the device-class definition with the names in the directory list for the associated path definition. The following illustrates the importance of matching device classes and paths to ensure that storage agents can access newly created FILE volumes.

Suppose you want to use these three directories for a FILE library:

```
/opt/tivoli1
/opt/tivoli2
```

## UPDATE PATH

/opt/tivoli3

1. You use the following command to set up a FILE library named CLASSA with one drive named *CLASSA1* on *SERVER1*:

```
define devclass classa devtype=file
directory="/opt/tivoli1,/opt/tivoli2,/opt/tivoli3"
shared=yes mountlimit=1
```

2. You want the storage agent STA1 to be able to use the FILE library, so you define the following path for storage agent STA1:

```
define path server1 sta1 srctype=server desttype=drive device=file
directory="/opt/ibm1,/opt/ibm2,/opt/ibm3" library=classa
```

In this scenario, the storage agent, STA1, will replace the directory name /opt/tivoli1 with the directory name /opt/ibm1/ to access FILE volumes that are in the /opt/tivoli1 directory on the server.

3. If file volume /opt/tivoli1/file1.dsm is created on *SERVER1*, and if the following command is issued,

```
update devclass classa directory="/opt/otherdir,/opt/tivoli2,
/opt/tivoli3"
```

*SERVER1* will still be able to access file volume /opt/tivoli1/file1.dsm, but the storage agent STA1 will not be able to access it because a matching directory name in the PATH directory list no longer exists. If a directory name is not available in the directory list associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume will still be accessible from the Tivoli Storage Manager server for reading, failure of the storage agent to access the FILE volume can cause operations to be retried on a LAN-only path or to fail.

### Example: Update a path from a data mover NAS file server to a tape drive

Update a path from a data mover that is a NAS file server to the drive *TAPEDRV2* that the data mover uses for backup and restore operations. In this example, the NAS data mover is *NAS1*, the library is *NASLIB*, and the device name for the drive is *rst01*.

```
update path nas1 tapedrv2 srctype=datamover desttype=drive library=naslib
device=rst01
```

### Related commands

Table 374. Commands related to UPDATE PATH

| Command          | Description   |
|------------------|---|
| DEFINE DATAMOVER | Defines a data mover to the IBM Tivoli Storage Manager server.      |
| DEFINE DRIVE     | Assigns a drive to a library.                                       |
| DEFINE LIBRARY   | Defines an automated or manual library.                             |
| DEFINE PATH      | Defines a path from a source to a destination.                      |
| DELETE PATH      | Deletes a path from a source to a destination.                      |
| QUERY PATH       | Displays information about the path from a source to a destination. |
| UPDATE DATAMOVER | Changes the definition for a data mover.                            |

## UPDATE POLICYSET (Update a policy set description)

Use this command to change the description of a policy set. You cannot change 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—domain_name—policy_set_name—————►
►—DESCription—==description—————►◄
```

### Parameters

#### *domain\_name* (Required)

Specifies the policy domain to which the policy set belongs.

#### *policy\_set\_name* (Required)

Specifies the policy set to update. You cannot change the ACTIVE policy set.

#### DESCription (Required)

Specifies text that describes the policy set. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove a previously defined description, specify a null string ("").

### Example: Update a policy set

Update a policy set called VACATION for the EMPLOYEE\_RECORDS policy domain with a description of "Schedule Planning Information."

```
update policyset employee_records vacation
description="schedule planning information"
```

### Related commands

Table 375. Commands related to UPDATE POLICYSET

| Command            | Description   |
|--------------------|---|
| ACTIVATE POLICYSET | Validates and activates a policy set.   |
| COPY MGMTCLASS     | Creates a copy of a management class.   |
| DEFINE DOMAIN      | Defines a policy domain that clients can be assigned to.                                      |
| DEFINE MGMTCLASS   | Defines a management class.   |
| DEFINE POLICYSET   | Defines a policy set within the specified policy domain.                                      |
| DELETE POLICYSET   | Deletes a policy set, including its management classes and copy groups, from a policy domain. |
| QUERY POLICYSET    | Displays information about policy sets.   |

## UPDATE POLICYSET

*Table 375. Commands related to UPDATE POLICYSET (continued)*

| Command            | Description  |
|--------------------|--|
| VALIDATE POLICYSET | Verifies and reports on conditions the administrator must consider before activating the policy set. |

## UPDATE PROFILE (Update a profile description)

Use this command on a configuration manager to update a profile description.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—UPDate PROFILE—*profile\_name*—DESCription—==—*description*—◄◄

### Parameters

*profile\_name* **(Required)**  
Specifies the profile to update.

**DESCription (Required)**  
Specifies a description for the profile. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove a description, specify a null string ("").

### Example: Update a profile's description

Update the description for profile DELTA.  
update profile delta description="PAYROLL domain"

### Related commands

Table 376. Commands related to UPDATE PROFILE

| Command                | Description  |
|------------------------|--|
| COPY PROFILE           | Creates a copy of a profile.                                       |
| DEFINE PROFASSOCIATION | Associates objects with a profile.                                 |
| DEFINE PROFILE         | Defines a profile for distributing information to managed servers. |
| DELETE PROFASSOCIATION | Deletes the association of an object with a profile.               |
| DELETE PROFILE         | Deletes a profile from a configuration manager.                    |
| LOCK PROFILE           | Prevents distribution of a configuration profile.                  |
| QUERY PROFILE          | Displays information about configuration profiles.                 |
| SET CONFIGMANAGER      | Specifies whether a server is a configuration manager.             |
| UNLOCK PROFILE         | Enables a locked profile to be distributed to managed servers.     |

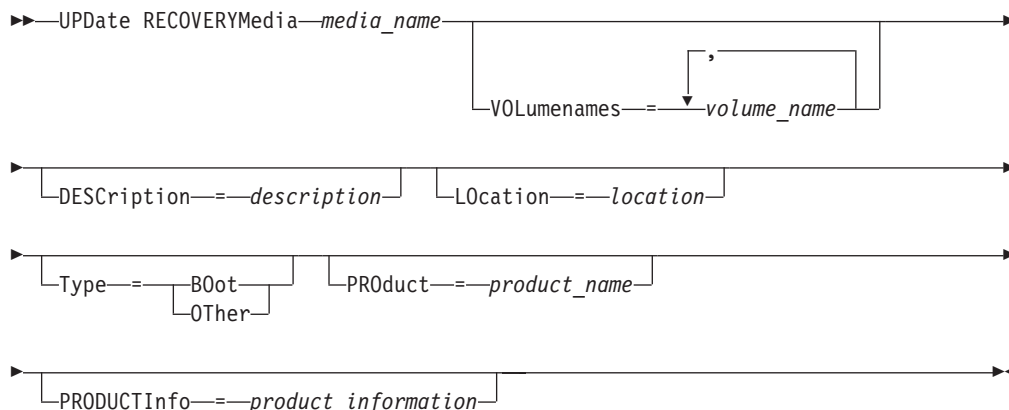
## UPDATE RECOVERYMEDIA (Update recovery media)

Use this command to update information about recovery media.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *media\_name* (Required)

Specifies the name of the recovery media to be updated.

#### VOLumenames

Specifies the names of volumes that contain the recoverable data (for example, operating system image copies). If you specify a `TYPE=BOOT`, you must specify the boot media volume names in the order in which they are to be loaded at recovery time. The volume names list can be up to 255 characters. Enclose the list in quotation marks if it contains any blank characters. To remove all volume names, specify a null string (`""`).

#### DESCription

Specifies the description of the recovery media. This parameter is optional. You can use up to 255 characters. Enclose the text in quotation marks if it contains any blank characters.

#### L0cation

Describes the location of the recovery media. This parameter is optional. You can use up to 255 characters. Enclose the text in quotation marks if it contains any blank characters. To remove a location description, specify a null string (`""`) for the value.

#### Type

Specifies the type of recovery media. This parameter is optional. Possible values are:

##### BOot

Specifies that this is boot media. You must specify volume names if the type is `BOOT`.

##### OTHer

Specifies that this is not boot media. For example, a CD that contains operating system manuals.

**PROduct**

Specifies the name of the product that wrote to this media. This parameter is optional. You can use up to 16 characters. Enclose the text in quotation marks if it contains any blank characters. To remove a product name, specify a null string ("" ) for the value.

**PRODUCTInfo**

Specifies any information about the product that wrote to the media that you may need to restore the machine. This parameter is optional. You can use up to 255 characters. Enclose the text in quotation marks if it contains any blank characters. To remove previously defined product information, specify a null string ("" ) for the value.

**Example: Update a recovery media's location description**

Update the location description for recovery media DIST5RM to "Corporate Headquarters Data Vault."

```
update recoverymedia dist5rm
location="Corporate Headquarters Data Vault"
```

**Related commands**

*Table 377. Commands related to UPDATE RECOVERYMEDIA*

| Command              | Description                                      |
|----------------------|--|
| DEFINE RECOVERYMEDIA | Defines the media required to recover a machine. |
| DELETE RECOVERYMEDIA | Deletes recovery media.                          |
| QUERY RECOVERYMEDIA  | Displays media available for machine recovery.   |

## UPDATE SCHEDULE (Update a schedule)

Use this command to update a client or administrative command schedule.

The UPDATE SCHEDULE command takes two forms, depending on whether the schedule applies to client operations or administrative commands. Within these two forms, you can select either classic or enhanced style schedules. The syntax and parameters for each form are defined separately.

*Table 378. Commands related to UPDATE SCHEDULE*

| Command              | Description   |
|----------------------|---|
| COPY SCHEDULE        | Creates a copy of a schedule.   |
| DEFINE SCHEDULE      | Defines a schedule for a client operation or an administrative command.                         |
| DELETE SCHEDULE      | Deletes a schedule from the database.   |
| QUERY EVENT          | Displays information about scheduled and completed events for selected clients.                 |
| QUERY SCHEDULE       | Displays information about schedules.   |
| SET MAXCMDRETRIES    | Specifies the maximum number of retries after a failed attempt to execute a scheduled command.  |
| SET MAXSCHEDSESSIONS | Specifies the maximum number of client/server sessions available for processing scheduled work. |
| SET RETRYPERIOD      | Specifies the time between retry attempts by the client scheduler.                              |



## UPDATE SCHEDULE (Update a client schedule)

Use the UPDATE SCHEDULE to update selected parameters for a client schedule.

This command does not change the client associations that have been made to this schedule. Any clients that are associated with the original schedule process the modified schedule.

Not all clients can run all scheduled operations, even though you can define the schedule on the server and associate it with the client. For example, a Macintosh client cannot run a schedule when the action is to restore or retrieve files, or run an executable script. An executable script is also known as a command file, a batch file, or a script on different client operating systems.

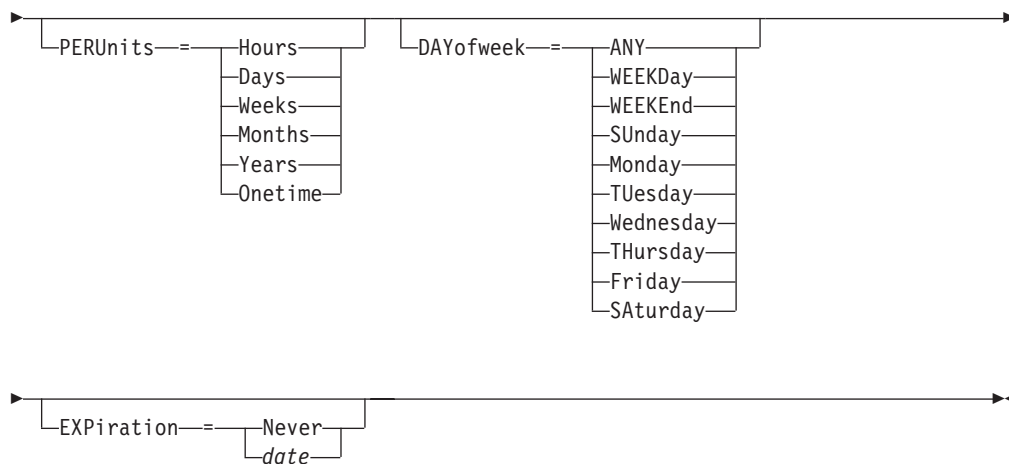
### Privilege class

To update a client schedule, you must have system privilege, unrestricted policy privilege, or restricted policy privilege for the policy domain to which the schedule belongs.

### Syntax for a classic client schedule



## UPDATE SCHEDULE

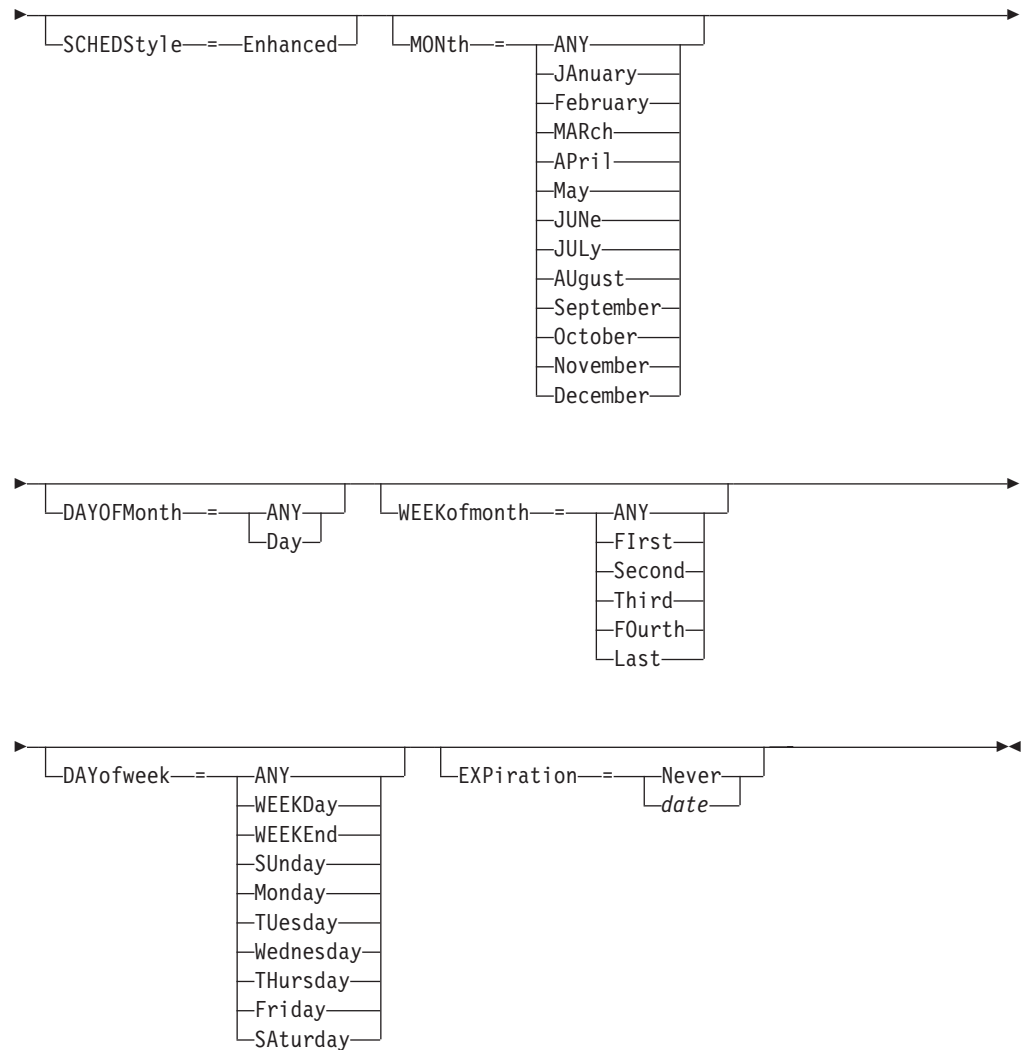


### Notes:

- 1 You must specify at least one optional parameter on this command.

### Syntax for an enhanced client schedule





## Notes:

- 1 You must specify at least one optional parameter on this command.

## Parameters

### *domain\_name* (Required)

Specifies the name of the policy domain to which this schedule belongs.

### *schedule\_name* (Required)

Specifies the name of the schedule to be updated.

### Type=Client

Specifies that a client schedule is updated. This parameter is optional. The default is CLIENT.

### DESCription

Specifies a description of the schedule. This parameter is optional. You can specify up to 255 characters for the description. Enclose the description in quotation marks if it contains blank characters. To remove a previously defined description, specify a null string ("" ) for this value.

## UPDATE SCHEDULE

### **ACTion**

Specifies the action that occurs when this schedule is processed. Possible values are:

#### **Incremental**

Specifies that the schedule backs up all files that are new or that have changed since the last incremental backup. Incremental also backs up any file for which all existing backups may have expired.

#### **Selective**

Specifies that the schedule backs up only files that are specified with the OBJECTS parameter.

#### **Archive**

Specifies that the schedule archives files that are specified with the OBJECTS parameter.

#### **REStore**

Specifies that the schedule restores files that are specified with the OBJECTS parameter.

When you specify ACTION=RESTORE for a scheduled operation, and the REPLACE option is set to PROMPT, no prompting occurs. If you set the option to PROMPT, the files are skipped.

If you specify a second file specification, this second file specification acts as the restore destination. If you need to restore multiple groups of files, schedule one for each file specification that you need to restore.

#### **REtrieve**

Specifies that the schedule retrieves files that are specified with the OBJECTS parameter.

#### **IMAGEBACKup**

Specifies that the schedule backs up logical volumes that are specified with the OBJECTS parameter.

#### **IMAGERESTore**

Specifies that the schedule restores logical volumes that are specified with the OBJECTS parameter.

#### **Command**

Specifies that the schedule processes a client operating system command or script that is specified with the OBJECTS parameter.

#### **Macro**

Specifies that a client processes a macro whose file name is specified with the OBJECTS parameter.

### **OPTions**

Specifies the client options that you specify to the scheduled command at the time the schedule is processed. This parameter is optional.

Only those options that are valid on the scheduled command may be specified for this parameter. Refer to the appropriate client manual for information about options that are valid from the command line. All options described there as valid only on the initial command line will either result in an error or simply be ignored when the schedule from the server is run. For example, do not include the following options because they have no effect when the client processes the scheduled command:

MAXCMDRETRIES

OPTFILE

QUERYSCHEDPERIOD  
 RETRYPERIOD  
 SCHEDLOGNAME  
 SCHEDMODE  
 SERVERNAME  
 TCPCLIENTADDRESS  
 TCPCLIENTPORT

If the option string contains multiple options or options with embedded spaces, surround the entire option string with one pair of single quotes. Individual options that contain spaces should be enclosed in double quotes. A leading minus sign is required in front of the option. Errors can occur if the option string contains spaces that are not quoted correctly.

The following examples show how to specify some client options:

- To specify `subdir=yes` and domain `all-local -systemobject`, enter:  
`options='-subdir=yes -domain="all-local -c: -systemobject"'`
- To specify domain `all-local -c: -d:`, enter:  
`options='-domain="all-local -c: -d:"'`

## OBjects

Specifies the objects for which the specified action is performed. Use a single space between each object. This parameter is required except when `ACTION=INCREMENTAL`. If the action is a backup, archive, retrieve, or restore operation, the objects are file spaces, directories, or logical volumes. See the *Backup-Archive Clients Installation and User's Guide* for command syntax information. If the action is to run a command or macro, the object is the name of the command or macro to run.

When you specify `ACTION=INCREMENTAL` without specifying a value for this parameter, the scheduled command is invoked without specified objects and attempts to process the objects as defined in the client's option file. To select all file spaces or directories for an action, explicitly list them in the object string. Entering only an asterisk in the object string causes the backup to occur only for the directory where the scheduler was started.

### Important:

- If you specify a second file specification, and it is not a valid destination, you receive this error:  
`ANS1082E Invalid destination file specification <filespec> entered.`
- If you specify more than two file specifications, you receive this error:  
`ANS1102E Excessive number of command line arguments passed to the program!`

When you specify `ACTION=ARCHIVE`, `INCREMENTAL`, or `SELECTIVE` for this parameter, you can list a maximum of twenty (20) file specifications.

Enclose the object string in double quotes if it contains blank characters (spaces), and then surround the double quotes with single quotes. If the object string contains multiple file names, enclose each file name with its own pair of double quotes, then surround the entire string with one pair of single quotes. Errors can occur if file names contain a space that is not quoted correctly. The following examples show how to specify some file names:

- To specify `/home/file 2`, `/home/gif files`, and `/home/my test file`, enter:  
`OBJECTS='" /home/file 2" " /home/gif files" " /home/my test file"'`

## UPDATE SCHEDULE

- To specify /home/test file, enter:  
OBJECTS='/home/test file'

### PRIority

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer from 1 to 10, with 1 being the highest priority and 10 being the lowest. The default is 5.

If two or more schedules have the same window start time, the value you specify determines when Tivoli Storage Manager processes the schedule. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

### STARTDate

Specifies the date for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current date. Use this parameter with the STARTTIME parameter to specify when the initial startup window of the schedule starts.

You can specify the date using one of the values below:

| Value               | Description   | Example         |
|---------------------|---|-----------------|
| MM/DD/YYYY          | A specific date   | 09/15/1998      |
| TODAY               | The current date  | TODAY           |
| TODAY+days or +days | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3. |

### STARTTime

Specifies the time for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current time. This parameter is used in conjunction with the STARTDATE parameter to specify when the initial startup window begins.

You can specify the time using one of the values below:

| Value               | Description  | Example   |
|---------------------|--|---|
| HH:MM:SS            | A specific time                                    | 10:30:08  |
| NOW                 | The current time                                   | NOW   |
| NOW+HH:MM or +HH:MM | The current time plus hours and minutes specified  | NOW+02:00 or +02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW+02:00 or STARTTIME=+02:00, the beginning of the startup window is at 7:00. |
| NOW-HH:MM or -HH:MM | The current time minus hours and minutes specified | NOW-02:00 or -02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW-02:00 or STARTTIME=-02:00, the beginning of the startup window is at 3:00. |

### DURation

Specifies the number of units that define the length of the startup window of the scheduled operation. This parameter is optional. This value must be from 1 to 999. The default is 1.

Use this parameter 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 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.

**Tip:** Define schedules with durations longer than 10 minutes. Doing this will give the Tivoli Storage Manager scheduler enough time to process the schedule and prompt the client.

### DURUnits

Specifies the time units used to determine the duration of the window in which the schedule can start. This parameter is optional. The default is HOURS.

Use this parameter with the DURATION parameter to specify how long the startup window remains open to process 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 schedule may not necessarily complete processing within this window. If the schedule needs to be retried for any reason, the retry attempts must begin before the startup window elapses, or the operation does not restart.

The default value for the length of the startup window is 1 hour. 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 of the scheduled operation 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 specify PERUNITS=ONETIME. The INDEFINITE value is not allowed with enhanced schedules.

### SCHEDStyle

This parameter is optional. SCHEDSTYLE defines either the interval between times when a schedule should run, or the days on which it should run. The style can be either **classic** or **enhanced**. This parameter must be specified when you change a schedule from classic to enhanced or back to classic. Otherwise, the value for the existing schedule is used.

For classic schedules, these parameters are allowed: PERIOD, PERUNITS, and DAYOFWEEK. These parameters are not allowed: MONTH, DAYOFMONTH, and WEEKOFMONTH. If the previous schedule style was enhanced, the MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK parameters will be reset. DAYOFWEEK, PERIOD, and PERUNITS will be set to default values unless they are specified with the update command.

For enhanced schedules, these parameters are allowed: MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK. These parameters are

## UPDATE SCHEDULE

not allowed: PERIOD and PERUNITS. If the previous schedule style was classic, the DAYOFWEEK, PERIOD, and PERUNITS parameters will be reset. MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK will be set to default values unless they are specified with the update command.

### **PERiod**

Specifies the length of time between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. You can specify an integer from 1 to 999. The default is 1.

Use this parameter 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 five days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default is 1 day.

This value is ignored if you specify PERUNITS=ONETIME.

### **PERUnits**

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. The default is DAYS.

Use this parameter 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 default is 1 day. Possible values are:

#### **Hours**

Specifies that the time between startup windows is in hours.

#### **Days**

Specifies that the time between startup windows is in days.

#### **Weeks**

Specifies that the time between startup windows is in weeks.

#### **Months**

Specifies that the time between startup windows is in months.

When you specify PERUNITS=MONTHS, the scheduled operation will be processed each month on the same date. For example, if the start date for the scheduled operation is 02/04/1998, the schedule will process on the 4th of every month thereafter. However, if the date is not valid for the next month, then the scheduled operation will be processed on the last valid date in the month. Thereafter, subsequent operations are based on this new date. For example, if the start date is 03/31/1998, the next month's operation will be scheduled for 04/30/1998. Thereafter, all subsequent operations will be on the 30th of the month until February. Because February has only 28 days, the operation will be scheduled for 02/28/1999. Subsequent operations will be processed on the 28th of the month.

#### **Years**

Specifies that the time between startup windows for the schedule is in years.

When you specify PERUNITS=YEARS, the scheduled operation will be processed on the same month and date of each year. For example, if the start date for the scheduled operation is 02/29/2004, the next year's



scheduled operation will be 02/28/2005 because February only has 28 days. Thereafter, subsequent operations will be scheduled for February 28th.

#### **Onetime**

Specifies that the schedule processes once. This value overrides the value you specified for the PERIOD parameter.

#### **DAYofweek**

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can specify different options for the DAYofweek parameter, depending on whether the schedule style has been defined as Classic or Enhanced:

##### **Classic Schedule**

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can either specify one day of the week, or WEEKDAY, WEEKEND, or ANY. If the start date and start time fall on a day that does not correspond to a day you specify, the start date and start time 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 processed when you would expect. The default is ANY.

##### **Enhanced Schedule**

Specifies the days of the week on which to run the schedule. You can either specify multiple days separated by commas and no intervening blanks, or WEEKDAY, WEEKEND, or ANY. If you specify multiple days, the schedule will run on each of the specified days. If you specify WEEKDAY or WEEKEND, you must also specify either WEEKOFMONTH=FIRST or WEEKOFMONTH=LAST, and the schedule will run just once per month.

The default value is ANY, meaning the schedule will run every day of the week or on the day or days determined by other enhanced schedule parameters. DAYOFWEEK must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

Possible values for the DAYofweek parameter 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.

## UPDATE SCHEDULE

### **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.

### **MONth**

Specifies the months of the year during which to run the schedule. This parameter is used only with enhanced schedules. Specify multiple values by using commas and no intervening blanks. The default value is ANY. This means the schedule will run during every month of the year.

### **DAYOFMonth**

Specifies the day of the month to run the schedule. This parameter is used only with enhanced schedules. You can either specify ANY or a number from -31 through 31, excluding zero. Negative values are a day from the end of the month, counting backwards. For example, the last day of the month is -1, the next-to-the-last day of the month is -2, etc. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run on each of the specified days of the month. If multiple values resolve to the same day, the schedule will run only once that day.

The default value is ANY. This means the schedule will run on every day of the month or on the days determined by other enhanced schedule parameters. DAYOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFWEEK or WEEKOFMONTH parameters.

If an existing schedule specifies a value other than ANY for DAYOFWEEK and WEEKOFMONTH, and DAYOFMONTH is updated, DAYOFWEEK and WEEKOFMONTH are reset to ANY.

### **WEEKofmonth**

Specifies the week of the month in which to run the schedule. This parameter is used only with enhanced schedules. A week is considered any seven-day period which does not start on a particular day of the week. You can specify FIRST, SECOND, THIRD, FOURTH, LAST, or ANY. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run during each of the specified weeks of the month. If multiple values resolve to the same week, the schedule will run only once during that week.

The default value is ANY, meaning the schedule will run during every week of the month or on the day or days determined by other enhanced schedule parameters. WEEKOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

### **EXPIration**

Specifies the date after which this schedule is no longer used. This parameter is optional. The default is NEVER. Possible values are:

#### **Never**

Specifies that the schedule never expires.

### *expiration\_date*

Specifies the date on which this schedule expires, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the date you specify.

### **Example: Update a schedule's priority**

Update the MONTHLY\_BACKUP schedule that belongs to the STANDARD policy domain by setting its priority value to 1.

```
update schedule standard monthly_backup priority=1
```

### **Example: Update a schedule's expiration date**

Update the WEEKLY\_BACKUP schedule that belongs to the EMPLOYEE\_RECORDS policy domain to expire on March 29, 1999 (03/29/1999).

```
update schedule employee_records weekly_backup expiration=03/29/1999
```

### **Example: Update a schedule to archive on the last Friday of a month**

Update a schedule from archiving files quarterly on the last Friday of the month to archiving on the last day of the specified months.

```
update schedule employee_records quarterly_archive dayofmonth=-1
```

WEEKOFMONTH and DAYOFWEEK will be reset to ANY.

## UPDATE SCHEDULE

### UPDATE SCHEDULE (Update an administrative schedule)

Use this command to update selected parameters for an administrative command schedule.

You cannot schedule MACRO or QUERY ACTLOG commands.

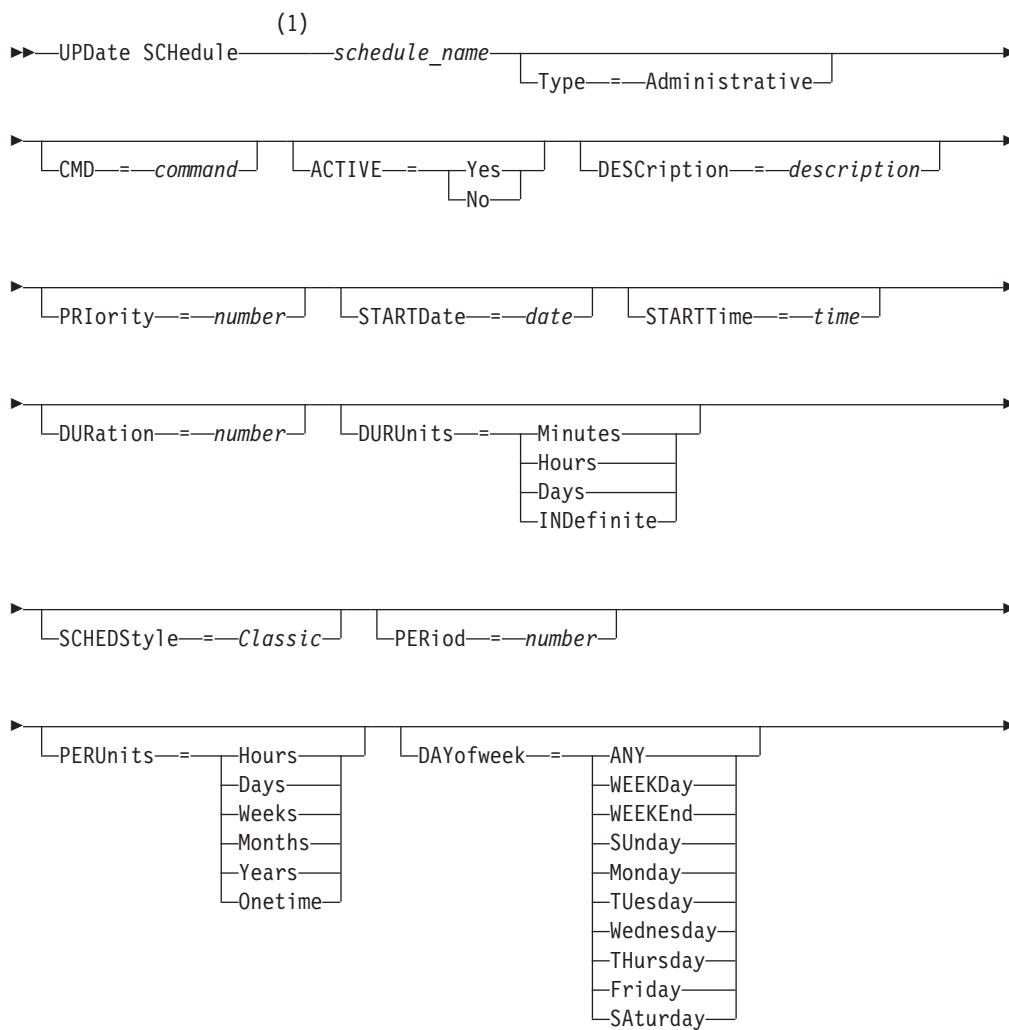
A managed administrative schedule that is updated by a configuration manager is set to an inactive state on the managed servers during configuration refresh processing. It remains in an inactive state until it is updated to an active state on those servers.

### Privilege class

To update an administrative schedule, you must have system privilege.

### Syntax

#### Classic administrative schedule



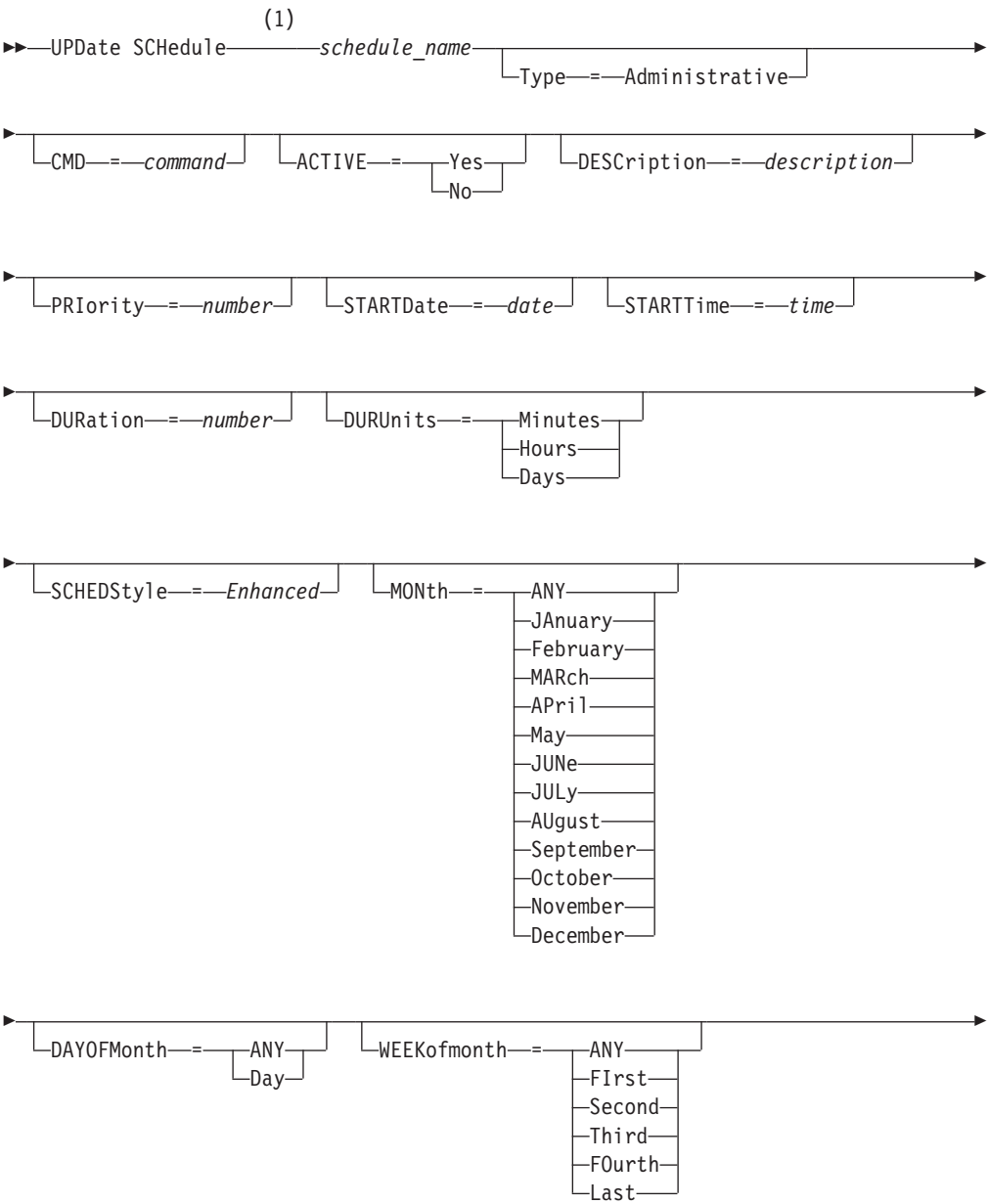


Notes:

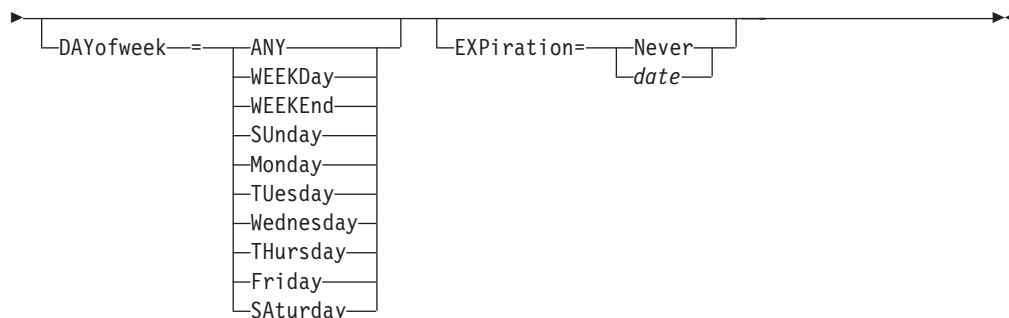
- 1 You must specify at least one optional parameter on this command.

Syntax

Enhanced administrative schedule



## UPDATE SCHEDULE



### Notes:

- 1 You must specify at least one optional parameter on this command.

### Parameters

#### *schedule\_name* (Required)

Specifies the name of the schedule to be updated.

#### **Type=Administrative** (Required)

Specifies that an administrative command schedule is updated.

#### **CMD**

Specifies the administrative command to be scheduled for processing. This parameter is optional. The command you specify can contain up to 512 characters. Enclose the command in quotation marks if it contains blanks.

You cannot specify redirection characters with this parameter.

#### **ACTIVE**

Specifies whether the administrative command is eligible for processing. This parameter is optional. An administrative command schedule will not be processed unless it is set to the active state. Possible values are:

##### **YES**

Specifies that the administrative command is eligible for processing.

##### **NO**

Specifies that the administrative command is not eligible for processing.

#### **DESCription**

Specifies a description of the schedule. This parameter is optional. You can specify up to 255 characters for the description. Enclose the description in quotation marks if it contains blanks. To remove a previously defined description, specify a null string ("" ) for this value.

#### **PRIority**

Specifies the priority value for a schedule. This parameter is optional. You can specify an integer from 1 to 10, with 1 being the highest priority and 10 being the lowest. The default is 5.

If two or more schedules have the same window start time, the value you specify determines when Tivoli Storage Manager processes the schedule. The schedule with the highest priority starts first. For example, a schedule with PRIORITY=3 starts before a schedule with PRIORITY=5.

#### **STARTDate**

Specifies the date for the beginning of the window in which the schedule is

first processed. This parameter is optional. The default is the current date. Use this parameter with the STARTTIME parameter to specify when the initial startup window of the schedule starts.

You can specify the date using one of the values below:

| Value                             | Description   | Example         |
|-----------------------------------|---|-----------------|
| <i>MM/DD/YYYY</i>                 | A specific date   | 09/15/1998      |
| <b>TODAY</b>                      | The current date  | TODAY           |
| <b>TODAY+days</b> or <i>+days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3. |

### STARTTime

Specifies the time for the beginning of the window in which the schedule is first processed. This parameter is optional. The default is the current time. This parameter is used in conjunction with the STARTDATE parameter to specify when the initial startup window begins.

You can specify the time using one of the values below:

| Value                             | Description  | Example   |
|-----------------------------------|--|---|
| <i>HH:MM:SS</i>                   | A specific time                                    | 10:30:08  |
| <b>NOW</b>                        | The current time                                   | NOW   |
| <b>NOW+HH:MM</b> or <i>+HH:MM</i> | The current time plus hours and minutes specified  | NOW+02:00 or +02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW+02:00 or STARTTIME=+02:00, the beginning of the startup window is at 7:00. |
| <b>NOW-HH:MM</b> or <i>-HH:MM</i> | The current time minus hours and minutes specified | NOW-02:00 or -02:00.<br><br>If you issue this command at 5:00 with STARTTIME=NOW-02:00 or STARTTIME=-02:00, the beginning of the startup window is at 3:00. |

### DURation

Specifies the number of units that define the length of the startup window of the scheduled operation. This parameter is optional. This value must be from 1 to 999. The default is 1.

Use this parameter 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 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

Specifies the time units used to determine the duration of the window in which the schedule can start. This parameter is optional. The default is HOURS.

## UPDATE SCHEDULE

Use this parameter with the DURATION parameter to specify how long the startup window remains open to process 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 schedule may not necessarily complete processing within this window. If the schedule needs to be retried for any reason, the retry attempts must begin before the startup window elapses, or the operation does not restart.

The default value for the length of the startup window is 1 hour. 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 of the scheduled operation 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 specify PERUNITS=ONETIME. The INDEFINITE value is not allowed with enhanced schedules.

## **SCHEDStyle**

This parameter is optional. SCHEDSTYLE defines either the interval between times when a schedule should run, or the days on which it should run. The style can be either **classic** or **enhanced**. This parameter must be specified when you change a schedule from classic to enhanced or back to classic. Otherwise, the value for the existing schedule is used.

For classic schedules, these parameters are allowed: PERIOD, PERUNITS, and DAYOFWEEK. These parameters are not allowed: MONTH, DAYOFMONTH, and WEEKOFMONTH. If the previous schedule style was enhanced, the MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK parameters will be reset. DAYOFWEEK, PERIOD, and PERUNITS will be set to default values unless they are specified with the update command.

For enhanced schedules, these parameters are allowed: MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK. These parameters are not allowed: PERIOD and PERUNITS. If the previous schedule style was classic, the DAYOFWEEK, PERIOD, and PERUNITS parameters will be reset. MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK will be set to default values unless they are specified with the update command.

## **PERiod**

Specifies the length of time between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. You can specify an integer from 1 to 999. The default is 1.

Use this parameter 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 five days after the initial start date and start time. The period between startup windows must exceed the duration of each window. The default is 1 day.

This value is ignored if you specify PERUNITS=ONETIME.



**PERUnits**

Specifies the time units used to determine the period between startup windows for this schedule. This parameter is optional. This parameter is used only with classic schedules. The default is DAYS.

Use this parameter 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 default is 1 day. Possible values are:

**Hours**

Specifies that the time between startup windows is in hours.

**Days**

Specifies that the time between startup windows is in days.

**Weeks**

Specifies that the time between startup windows is in weeks.

**Months**

Specifies that the time between startup windows is in months.

When you specify PERUNITS=MONTHS, the scheduled operation will be processed each month on the same date. For example, if the start date for the scheduled operation is 02/04/1998, the schedule will process on the 4th of every month thereafter. However, if the date is not valid for the next month, then the scheduled operation will be processed on the last valid date in the month. Thereafter, subsequent operations are based on this new date. For example, if the start date is 03/31/1998, the next month's operation will be scheduled for 04/30/1998. Thereafter, all subsequent operations will be on the 30th of the month until February. Because February has only 28 days, the operation will be scheduled for 02/28/1999. Subsequent operations will be processed on the 28th of the month.

**Years**

Specifies that the time between startup windows for the schedule is in years.

When you specify PERUNITS=YEARS, the scheduled operation will be processed on the same month and date of each year. For example, if the start date for the scheduled operation is 02/29/2004, the next year's scheduled operation will be 02/28/2005 because February only has 28 days. Thereafter, subsequent operations will be scheduled for February 28th.

**Onetime**

Specifies that the schedule processes once. This value overrides the value you specified for the PERIOD parameter.

**DAYofweek**

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can specify different options for the DAYofweek parameter, depending on whether the schedule style has been defined as Classic or Enhanced:

**Classic Schedule**

Specifies the day of the week on which the startup window for the schedule begins. This parameter is optional. You can either specify one day of the week, or WEEKDAY, WEEKEND, or ANY. If the start date and start time fall on a day that does not correspond to a day you

specify, the start date and start time 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 processed when you would expect. The default is ANY.

### **Enhanced Schedule**

Specifies the days of the week on which to run the schedule. You can either specify multiple days separated by commas and no intervening blanks, or WEEKDAY, WEEKEND, or ANY. If you specify multiple days, the schedule will run on each of the specified days. If you specify WEEKDAY or WEEKEND, you must also specify either WEEKOFMONTH=FIRST or WEEKOFMONTH=LAST, and the schedule will run just once per month.

The default value is ANY, meaning the schedule will run every day of the week or on the day or days determined by other enhanced schedule parameters. DAYOFWEEK must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

Possible values for the DAYofweek parameter 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.

#### **Saturday**

Specifies that the startup window begins on Saturday.

### **MONTH**

Specifies the months of the year during which to run the schedule. This parameter is used only with enhanced schedules. Specify multiple values by using commas and no intervening blanks. The default value is ANY. This means the schedule will run during every month of the year.

### **DAYOFMonth**

Specifies the day of the month to run the schedule. This parameter can only be specified with enhanced schedules. You can either specify ANY or a number

from -31 through 31, excluding zero. Negative values are a day from the end of the month, counting backwards. For example, the last day of the month is -1, the next-to-the-last day of the month is -2, etc. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run on each of the specified days of the month. If multiple values resolve to the same day, the schedule will run only once that day.

The default value is ANY. This means the schedule will run on every day of the month or on the days determined by other enhanced schedule parameters. DAYOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFWEEK or WEEKOFMONTH parameters.

#### **WEEKofmonth**

Specifies the week of the month in which to run the schedule. This parameter can only be specified with enhanced schedules. A week is considered any seven-day period which does not start on a particular day of the week. You can specify FIRST, SECOND, THIRD, FOURTH, LAST, or ANY. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule will run during each of the specified weeks of the month. If multiple values resolve to the same week, the schedule will run only once during that week.

The default value is ANY, meaning the schedule will run during every week of the month or on the day or days determined by other enhanced schedule parameters. WEEKOFMONTH must have a value of ANY (either by default or specified with the command) when used with the DAYOFMONTH parameter.

#### **EXPIRATION**

Specifies the date after which this schedule is no longer used. This parameter is optional. The default is NEVER. Possible values are:

##### **Never**

Specifies that the schedule never expires.

##### *expiration\_date*

Specifies the date on which this schedule expires, in MM/DD/YYYY format. If you specify an expiration date, the schedule expires at 23:59:59 on the date you specify.

#### **Example: Update a backup schedule to every three days**

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.

```
update schedule backup_backuppool type=administrative cmd="backup stgpool
  backuppool copystg" active=yes starttime=22:00 period=3
```

#### **Example: Update a backup schedule to every first and third Friday**

Update a schedule named BACKUP\_ARCHIVEPOOL that backs up the primary storage pool ARCHIVEPOOL to the copy storage pool RECOVERYPOOL. The existing schedule runs on the first and tenth day of every month. Update it to run the first and third Friday of every month.

```
update schedule backup_archivepool
  dayofweek=friday weekofmonth=first,third
```

DAYOFMONTH will be reset to ANY.

### UPDATE SCRIPT (Update a Tivoli Storage Manager script)

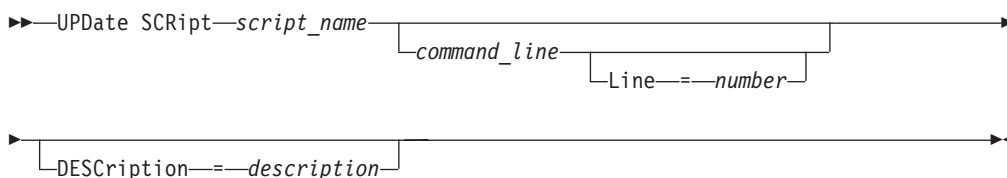
Use this command to change a command line or to add a new command line to an IBM Tivoli Storage Manager script.

The Administration Center only supports ASCII characters for input. If you need to enter characters that are not ASCII, issue the DEFINE SCRIPT and UPDATE SCRIPT commands from the server console.

#### Privilege class

To issue this command, the administrator must have previously defined the script or must have system privilege.

#### Syntax



#### Parameters

##### *script\_name* (Required)

Specifies the name of the script to be updated.

##### *command\_line*

Specifies a new or updated command to be processed in a script. You must update a command, a description, or both when you issue this command.

Command can contain substitution variables and may be continued across multiple lines if you specify a continuation character (-) as the last character in the command. You can specify up to 1200 characters for the command. Enclose the command in quotation marks if it contains blanks. If you specify this parameter, you can optionally specify the following parameter.

##### **Line**

Specifies the line number for the command. If you do not specify a line number, the command line is appended to the existing series of command lines. The appended command line is assigned a line number of five greater than the last command line number in the sequence. For example, if the last line in your script is 015, the appended command line is assigned a line number of 020.

If you specify a line number, the command will replace an existing line (if the number is the same as an existing line). Or the command will insert the specified line (if the line number does not correspond to an existing line number for the command line sequence).

##### **DESCRiption**

Specifies a description for the script. You can specify up to 255 characters for the description. Enclose the description in quotation marks if it contains blank characters.

### Example: Add a command to the end of a script

Assume that you have defined the following three line script, named QSAMPLE, and that you want to add the QUERY SESSION command to the end of the script.

```
001  /* This is a sample script */
005  QUERY STATUS
010  QUERY PROCESS
update script qsampl "query session"
```

After the command processes, the script now consists of the following lines:

```
001  /* This is a sample script */
005  QUERY STATUS
010  QUERY PROCESS
015  QUERY SESSION
```

### Example: Update a specific line a script

Using the script from the prior example, change line 010 so that it processes the QUERY STGPPOOL command instead of the QUERY PROCESS command:

```
update script qsampl "query stgppl" line=010
```

After the command processes, the script now consists of the following lines:

```
001  /* This is a sample script */
005  QUERY STATUS
010  QUERY STGPPOOL
015  QUERY SESSION
```

### Example: Insert a command in the middle of a script

Using the script from the prior example, insert a new command line (SET REGISTRATION OPEN) after the QUERY STATUS command line in the QSAMPLE script:

```
update script qsampl "set registration open"
line=007
```

After the command processes, the script now consists of the following lines:

```
001  /* This is a sample script */
005  QUERY STATUS
007  SET REGISTRATION OPEN
010  QUERY STGPPOOL
015  QUERY SESSION
```

## Related commands

Table 379. Commands related to UPDATE SCRIPT

| Command       | Description  |
|---------------|--|
| COPY SCRIPT   | Creates a copy of a script.                                |
| DEFINE SCRIPT | Defines a script to the IBM Tivoli Storage Manager server. |
| DELETE SCRIPT | Deletes the script or individual lines from the script.    |
| QUERY SCRIPT  | Displays information about scripts.                        |
| RENAME SCRIPT | Renames a script to a new name.                            |
| RUN           | Runs a script.   |

## UPDATE SERVER (Update a server defined for server-to-server communications)

Use this command to update a server definition.

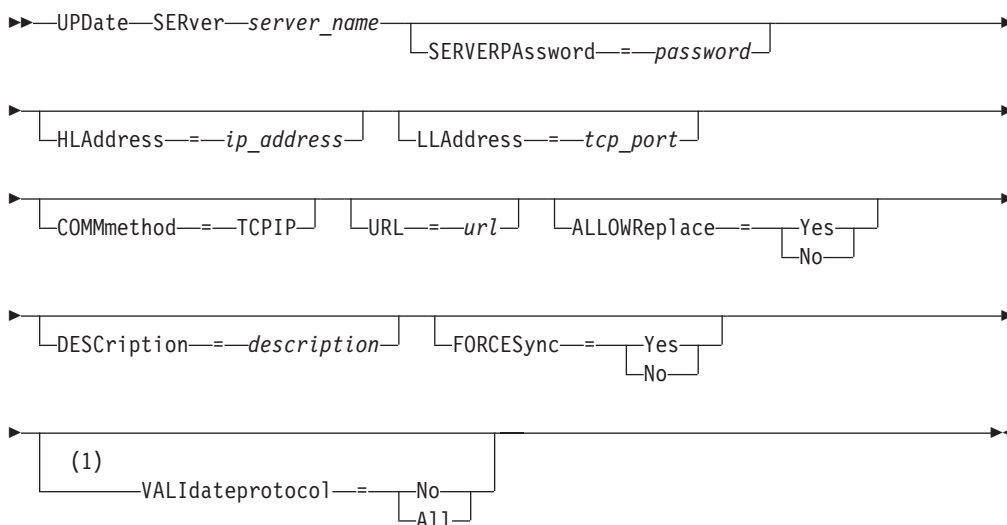
**Important:** If this is a source server for a virtual volume operation, changing any of these values can affect the ability of the source server to access and manage the data it has stored on the corresponding target server. Changing the server name using the SET SERVERNAME command might have additional implications, varying by platform. The following are some examples:

- Passwords might be invalidated
- Device information might be affected
- Registry information on Windows platforms might change

### Privilege class

To issue this command, you must have system privilege.

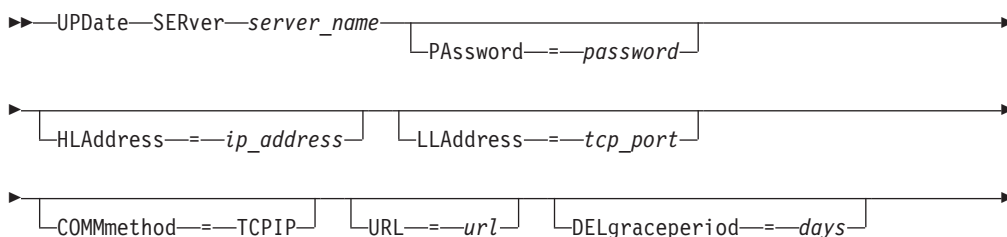
### Syntax for enterprise configuration, enterprise event logging, command routing, and storage agent

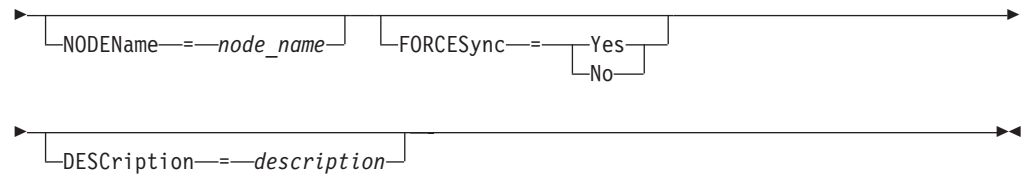


#### Notes:

- 1 The `VALIDATEPROTOCOL` parameter only applies to storage agent definitions.

### Syntax for virtual volumes





## Parameters

### *server\_name* (Required)

Specifies the name of the server to be updated. This parameter is required.

### **PAssword**

Specifies the password used to sign on to the target server for virtual volumes. This parameter is optional.

### **SERVERPAssword**

Specifies the server password, which is used for enterprise configuration, command routing, and server-to-server event logging functions. The password must match the server password set by the SET SERVERPASSWORD command. This parameter is optional.

### **HLAddress**

Specifies the IP address (in dotted decimal format) of the server. This parameter is optional.

### **LLAddress**

Specifies the low-level address of the server. This address is usually the same as that in the TCPPORT server option of the target server. This parameter is optional.

### **COMMmethod**

Specifies the communication method used to connect to the server. This parameter is optional.

### **URL**

Specifies the URL address used to access this server from a Web administrative interface. The parameter is optional.

### **DELgraceperiod**

Specifies a number of days that an object remains on the target server after it has been marked for deletion. Possible values are 0-9999. This parameter is optional.

### **NODENAME**

Specifies a node name to be used by the server to connect to the target server. This parameter is optional.

### **DESCription**

Specifies a description of the server. This parameter is optional. The description can be up to 255 characters. Enclose the description in quotation marks if it contains blank characters. To remove an existing description, specify a null string ("").

### **FORCESync**

Specifies whether to reset the server verification key when the source server next signs on to the target server. A valid verification key lets a source server put objects on the target server, manage the grace deletion period value, and update the password (if the current password is known and the verification key matches). The parameter is optional. Possible values are:

## UPDATE SERVER

### Yes

Specifies that a new verification key will be sent to and accepted by the target server if a valid password is received.

### No

Specifies that a new verification key will not be sent to the target server.

### VALIDateprotocol

Specify whether a cyclic redundancy check should be performed to validate the data sent between the storage agent and Tivoli Storage Manager server. The parameter is optional. The default is NO. Possible values are:

### No

Specifies that data validation not be performed on any data sent between the storage agent and server.

### All

Specifies that data validation be performed on all client file data, client file metadata, and the Tivoli Storage Manager server metadata that is sent between the storage agent and server. This mode impacts performance as additional overhead is required to calculate and compare CRC values between the storage agent and the server.

### ALLOWReplace

Specifies whether a server definition that was defined by a managed server can be replaced with a definition from the configuration manager. This parameter is optional. Possible values are:

### Yes

Specifies that a server definition can be replaced by a definition from the configuration manager.

### No

Specifies that a server definition cannot be replaced by the definition from the configuration manager.

## Example: Update a server's deletion grace period

Update the definition of SERVER2 to specify that objects remain on the target server for 10 days after they have been marked for deletion.

```
update server server2 delgraceperiod=10
```

## Example: Update a server's URL

Update the definition of NEWSERVER to specify its URL address to be `http://newserver:1580/`.

```
update server newserver url=http://newserver:1580/
```

## Related commands

*Table 380. Commands related to UPDATE SERVER*

| Command          | Description   |
|------------------|---|
| DEFINE DEVCLASS  | Defines a device class.                               |
| DEFINE SERVER    | Defines a server for server-to-server communications. |
| DELETE DEVCLASS  | Deletes a device class name.                          |
| DELETE FILESPACE | Deletes data associated with client's file spaces.    |



*Table 380. Commands related to UPDATE SERVER (continued)*

| Command           | Description  |
|-------------------|--|
| DELETE SERVER     | Deletes the definition of a server.  |
| QUERY NODE        | Displays partial or complete information about one or more clients.                    |
| QUERY SERVER      | Displays information about servers.  |
| RECONCILE VOLUMES | Reconciles source server virtual volume definitions and target server archive objects. |
| REGISTER NODE     | Defines a client to the server and sets options for that user.                         |
| REMOVE NODE       | Removes a client from the list of registered nodes for a specific policy domain.       |
| UPDATE DEVCLASS   | Changes the attributes of a device class.  |
| UPDATE NODE       | Changes the attributes associated with a client node.                                  |

## UPDATE SERVERGROUP (Update a server group description)

Use this command to update the description of a server group.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—UPDate SERVERGRoup—group_name—DESCription—==—description—◄◄
```

### Parameters

#### *group\_name* (Required)

Specifies the server group to update.

#### DESCription (Required)

Specifies a description of the server group. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains blank characters.

### Example: Update the description of a sever group

Update the description of the server group named WEST\_COMPLEX to "Western Region Complex".

```
update servergroup west_complex
description="western region complex"
```

### Related commands

*Table 381. Commands related to UPDATE SERVERGROUP*

| Command            | Description                               |
|--------------------|---|
| COPY SERVERGROUP   | Creates a copy of a server group.         |
| DEFINE SERVERGROUP | Defines a new server group.               |
| DELETE SERVERGROUP | Deletes a server group.                   |
| QUERY SERVERGROUP  | Displays information about server groups. |
| RENAME SERVERGROUP | Renames a server group.                   |

## UPDATE SPACETRIGGER (Update the space triggers)

Use this command to update settings for triggers that determine when and how the server resolves space shortages in storage pools that use sequential-access FILE and random-access DISK device classes.

For storage pools with a parameter RECLAMATIONTYPE=SNAPLOCK, space triggers are not enabled.

**Important:** Space trigger functions and storage pool space calculations take into account the space remaining in each directory. Ideally, you associate each directory with a separate file system. If you specify multiple directories for a device class and the directories reside in the same file system, the server calculates space by adding values representing the space remaining in each directory. These space calculations will be inaccurate. Rather than choosing a storage pool with sufficient space for an operation, the server might choose the wrong storage pool and run out of space prematurely. For space triggers, an inaccurate calculation might result in a failure to expand the space available in a storage pool. Failure to expand space in a storage pool is one of the conditions that can cause a trigger to become disabled. If a trigger is disabled because the space in a storage pool could not be expanded, you can re-enable the trigger by specifying the following command: update spacetrigger stg. No further changes are required to the space trigger.

See the DEFINE SPACETRIGGER command for more information.

### Privilege class

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

### Syntax

```

►►—UPDate SPACETrigger—STG—┐
                             └Fullpct—==—percent┘
►┐
  └SPACExpansion—==—percent┘ └EXPansionprefix—==—prefix┘
►┐
  └STGPOOL—==—storage_pool_name┘
  
```

### Parameters

#### STG (Required)

Specifies a storage pool space trigger

#### Fullpct

This parameter specifies the utilization percentage of the storage pool.

When this value is exceeded, the space trigger creates new volumes.

You can determine storage pool utilization by issuing the QUERY STGPOOL command with FORMAT=DETAILED. The percentage of storage pool utilization for the storage pool is displayed in the field "Space Trigger Util." The calculation for this percentage does not include potential scratch volumes. The calculation for the percentage utilization used for migration and reclamation, however, does include potential scratch volumes.

## UPDATE SPACETRIGGER

### SPACEexpansion

For space triggers for sequential-access FILE-type storage pools, this parameter is used in determining the number of additional volumes that are created in the storage pool. Volumes are created using the MAXCAPACITY value from the storage pool's device class. For space triggers for random-access DISK storage pools, the space trigger creates a single volume using the EXPANSIONPREFIX.

### EXPansionprefix

This specifies the prefix that the server uses to create new storage pool files. This parameter is optional and applies only to random-access DISK device classes. The default prefix is the server installation path.

The prefix can include one or more directory separator characters, for example:  
`/opt/tivoli/tsm/server/bin/`

You can specify up to 250 characters. If you specify a prefix that is not valid, automatic expansion can fail.

This parameter is not valid for space triggers for sequential-access FILE storage pools. Prefixes are obtained from the directories specified with the associated device class.

### STGPOOL

Specifies the storage pool associated with this space trigger. If the STGPOOL parameter is not specified, the default storage pool space trigger is updated.

This parameter does not apply to storage pools with the parameter RECLAMATIONTYPE=SNAPLOCK.

## Example: Increase the amount of space for a storage pool

Increase the amount of space in a storage pool by 50 percent when it is filled to 80 percent utilization of existing volumes. Space will be created in the directories associated with the device class.

```
update spacetrigger stg spaceexpansion=50 stgpool=file
```

## Related commands

*Table 382. Commands related to UPDATE SPACETRIGGER*

| Command             | Description   |
|---------------------|---|
| DEFINE SPACETRIGGER | Defines a space trigger to expand the space for a storage pool. |
| DELETE SPACETRIGGER | Deletes the storage pool space trigger.                         |
| QUERY SPACETRIGGER  | Displays information about a storage pool space trigger.        |

## UPDATE STGPOOL (Update a storage pool)

Use this command to change a storage pool.

The UPDATE STGPOOL command takes four forms:

- Updating a primary storage pool assigned to random access devices
- Updating a primary storage pool assigned to sequential access devices
- Updating a copy storage pool (always assigned to sequential access devices)
- Updating an active-data pool (always assigned to sequential access devices)

The syntax and parameters for each form are defined separately.

*Table 383. Commands related to UPDATE STGPOOL*

| Command                   | Description   |
|---------------------------|---|
| BACKUP STGPOOL            | Backs up a primary storage pool to a copy storage pool.                                       |
| COPY ACTIVATEDATA         | Copies active backup data.  |
| DEFINE COLLOCGROUP        | Defines a collocation group.  |
| DEFINE COLLOCMEMBER       | Adds a client node to a collocation group.  |
| DEFINE STGPOOL            | Defines a storage pool as a named collection of server storage media.                         |
| DELETE COLLOCGROUP        | Deletes a collocation group.  |
| DELETE COLLOCMEMBER       | Deletes a client node from a collocation group.   |
| DELETE STGPOOL            | Deletes a storage pool from server storage.   |
| MOVE DRMEDIA              | Moves DRM media on-site and off-site.   |
| MOVE MEDIA                | Moves storage pool volumes that are managed by an automated library.                          |
| QUERY COLLOCGROUP         | Displays information about collocation groups.  |
| QUERY DRMEDIA             | Displays information about disaster recovery volumes.   |
| QUERY NODEDATA            | Displays information about the location and size of data for a client node.                   |
| QUERY SHREDSTATUS         | Displays information about data waiting to be shredded.                                       |
| QUERY STGPOOL             | Displays information about storage pools.   |
| RESTORE STGPOOL           | Restores files to a primary storage pool from copy storage pools.                             |
| RESTORE VOLUME            | Restores files stored on specified volumes in a primary storage pool from copy storage pools. |
| SET DRMDBBACKUPEXPIREDAYS | Specifies criteria for database backup series expiration.                                     |
| SHRED DATA                | Manually starts the process of shredding deleted data.  |
| UPDATE COLLOCGROUP        | Updates the description of a collocation group.   |

## UPDATE STGPOOL

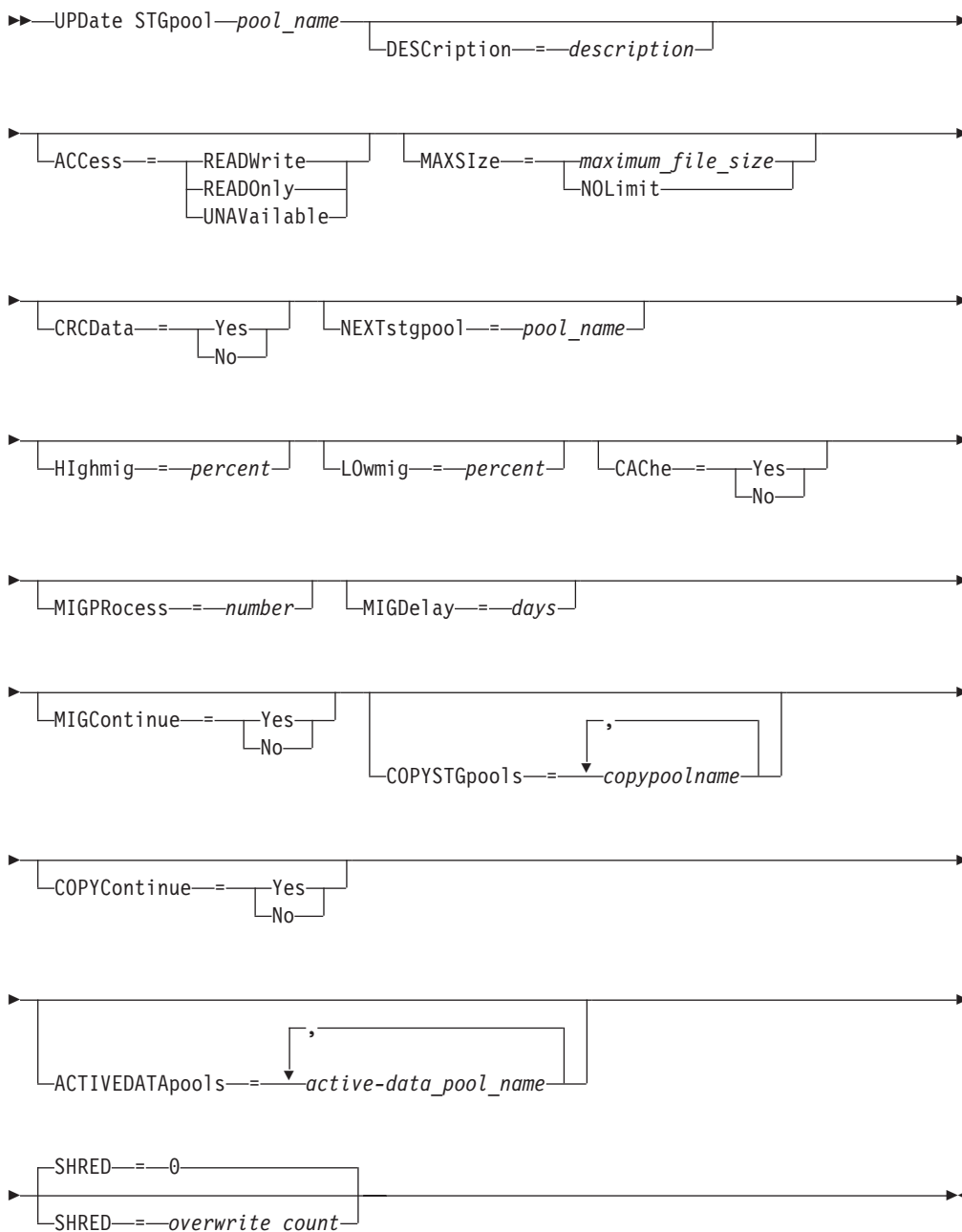
### UPDATE STGPOOL (Update a primary random access storage pool)

Use this command to update a random access storage pool.

#### 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



**Parameters*****pool\_name* (Required)**

Specifies the storage pool to update. This parameter is required.

**DESCRiption**

Specifies a description of the storage pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove an existing description, specify a null string ("").

**ACCess**

Specifies how client nodes and server processes (such as migration and reclamation) can access files in the storage pool. This parameter is optional. Possible values are:

**READWrite**

Specifies that client nodes and server processes can read and write to files stored on volumes in the storage pool.

**READOnly**

Specifies that client nodes can only read files from the volumes in the storage pool.

Server processes can move files within the volumes in the storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside 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 server processes attempt to write files to the storage pool.

**UNAVailable**

Specifies that client nodes cannot access files stored on volumes in the storage pool.

Server processes can move files within the volumes in the storage pool and can also move or copy files from this storage pool to another storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside the 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 server processes attempt to write files to the storage pool.

**MAXSIze**

Specifies the maximum size for a physical file that the server can store in the storage pool. This parameter is optional. Possible values are:

**NOLimit**

Specifies that there is no maximum size limit for physical files stored in the storage pool.

***maximum\_file\_size***

Limits the maximum physical file size. Specify an integer from 1 to 999999, followed by a scale factor. For example, MAXSIZE=5G specifies that the maximum file size for this storage pool is 5 gigabytes. Scale factors are:

**Scale factor    Meaning**

K    kilobyte

M    megabyte

## UPDATE STGPOOL

| Scale factor | Meaning |
|--------------|---------|
|--------------|---------|

|   |          |
|---|----------|
| G | gigabyte |
|---|----------|

|   |          |
|---|----------|
| T | terabyte |
|---|----------|

If a file exceeds the maximum size and no pool is specified as the next storage pool in the hierarchy, the server does not store the file. If a file exceeds the maximum size and a pool is specified as the next storage pool, the server stores the file in the next storage pool that can accept the file size. If you specify the next storage pool parameter, at least one storage pool in your hierarchy should have no limit on the maximum size of a file. By having no limit on the size for at least one pool, you ensure that no matter what its size, the server can store the file.

For logical files that are part of an aggregate, the server considers the size of the aggregate to be the file size. Therefore, the server does not store logical files that are smaller than the maximum size limit if the files are part of an aggregate that is larger than the maximum size limit.

### CRCData

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is optional. The default value is NO. By setting CRCData to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

#### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

#### No

Specifies that data is stored without CRC information.

### NEXTstgpool

Specifies a primary storage pool to which files are migrated. This parameter is optional.

To remove an existing storage pool from the storage hierarchy, specify a null string ("" ) for this value.

If this storage pool does not have a next storage pool, the server cannot migrate files from this storage pool and cannot store files that exceed the maximum size for this storage pool 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.

### Hlghmig

Specifies that the server starts migration for this storage pool when the amount of data in the pool reaches this percentage of the pool's estimated capacity. This parameter is optional. You can specify an integer from 0 to 100.

When the storage pool exceeds the high migration threshold, the server can start migration of files 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.



**LOWmig**

Specifies that the server stops migration for this storage pool when the amount of data in the pool reaches this percentage of the pool's estimated capacity. This parameter is optional. You can specify an integer from 0 to 99.

When the storage pool reaches the low migration threshold, the server does not start 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**

Specifies whether the migration process leaves a cached copy of a file in this storage pool after migrating the file to the next storage pool. This parameter is optional. Possible values are:

**Yes**

Specifies that caching is enabled.

**No**

Specifies that caching is disabled.

Using cache may improve the retrievability of files, but may affect the performance of other processes. See the *Administrator's Guide* for details.

**MIGPRocess**

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 to 999.

During migration, these processes are performed in parallel to provide the potential for improved migration rates.

**Remember:** The number of migration processes is dependent upon the setting of the MIGPROCESS parameter and the number of nodes or the number of collocation groups with data in the migrating storage pool. For example, if the MIGPROCESS parameter is equal to six, but there are only two nodes with data on the storage pool, migration processing only consists of two processes, not six.

**MIGDelay**

Specifies the minimum number of days a file must remain in a storage pool before it becomes eligible for migration. To calculate a value to compare to the specified MIGDELAY value, the server counts the number of days that the file has been in the storage pool and the number of days, if any, since the file was retrieved by a client. The lesser of the two values is compared to the specified MIGDELAY value. For example, if all the following conditions are true, a file is not migrated:

- A file has been in a storage pool for five days.
- The file was accessed by a client within the past three days.
- The value specified for the MIGDELAY parameter is four days.

This parameter is optional. You can specify an integer from 0 to 9999.

If you want the server to count the number of days based only on when a file was stored and not when it was retrieved, use the NORETRIEVEDATE server option.

### MIGContinue

Specifies whether you allow the server to migrate files that do not satisfy the migration delay time. This parameter is optional.

Because you can require that files remain in the storage pool for a minimum number of days, the server may migrate all eligible files to the next storage pool yet not meet the low migration threshold. This parameter allows you to specify whether the server is allowed to continue the migration process by migrating files that do not satisfy the migration delay time.

Possible values are:

#### Yes

Specifies that, when necessary to meet the low migration threshold, the server continues to migrate files that do not satisfy the migration delay time.

If you allow more than one migration process for the storage pool, some files that do not satisfy the migration delay time may be migrated unnecessarily. As one process migrates files that satisfy the migration delay time, a second process could begin migrating files that do not satisfy the migration delay time to meet the low migration threshold. The first process that is still migrating files that satisfy the migration delay time might have, by itself, caused the low migration threshold to be met.

#### No

Specifies that the server stops migration when no eligible files remain to be migrated, even before reaching the low migration threshold. The server does not migrate files unless the files satisfy the migration delay time.

### COPYSTGPools

Specifies the names of copy storage pools where the server simultaneously writes data. Beginning with Tivoli Storage Manager Version 5.3, you can specify a maximum of three copy pool names separated by commas. (In versions earlier than Version 5.3, the maximum number was ten.) Spaces between the names of the copy pools are not permitted. To add or remove one or more copy storage pools, specify the pool name or names that you want to include in the updated list. For example, if the existing copy pool list includes COPY1 and COPY2 and you want to add COPY3, specify `COPYSTGPools=COPY1,COPY2,COPY3`. To remove all existing copy storage pools associated with the primary storage pool, specify a null string ("" ) for the value (for example, `COPYSTGPools=""`).

When specifying a value for the COPYSTGPools parameter, you can also specify a value for the COPYCONTINUE parameter. For additional information, see the COPYCONTINUE parameter.

The combined total number of storage pools specified in the COPYSTGPools and ACTIVEDATAPOOLS parameters cannot exceed three.

The simultaneous write function changed in Tivoli Storage Manager Version 5.3. In Version 5.3 and later, when a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of copy storage pools and the COPYCONTINUE value from the primary storage pool. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to copy storage pools for the following operations:

- Backup and archive operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API
- Migration operations by Tivoli Storage Manager for Space Management clients
- Import operations that involve copying exported file data from external media to a primary storage pool associated with a copy storage pool list

**Restrictions:** Simultaneous write is not supported for the following store operations:

- When the operation is using LAN-free data movement. Simultaneous write takes precedence over LAN-free operations, causing the operations to go over the LAN. However, the simultaneous write configuration is honored.
- NAS backup operations. If the primary storage pool specified in the DESTINATION or TOCDESTINATION in the copy group of the management class has copy storage pools defined, the copy storage pools are ignored and the data is stored into the primary storage pool only.

**Attention:** The function provided by the COPYSTGPOLLS parameter is not intended to replace the BACKUP STGPOOL command. If you use the COPYSTGPOLLS parameter, continue to use the BACKUP STGPOOL command to ensure that the copy storage pools are complete copies of the primary storage pool. There are cases when a copy might not be created. For more information, see the COPYCONTINUE parameter description.

### **COPYContinue**

Specifies how the server should react to a copy storage pool write failure for any of the copy storage pools listed in the COPYSTGPOLLS parameter. This parameter is optional. When specifying the COPYCONTINUE parameter, either a COPYSTGPOLLS list must already exist or the COPYSTGPOLLS parameter must also be specified.

This parameter has no effect on active-data pools. If a write error occurs while writing to an active-data pool, the server will stop writing to the failing active-data pool for the remainder of the session.

Possible values are:

#### **Yes**

If the COPYCONTINUE option is set to YES, the server will stop writing to the failing copy pools for the remainder of the session, but continue storing files into the primary pool and any remaining copy pools. The copy storage pool list is active only for the life of the client session and applies to all the primary storage pools in a particular storage pool hierarchy.

For additional information about the COPYCONTINUE parameter, refer to the information about simultaneous write in the *Administrator's Guide*.

#### **No**

If the COPYCONTINUE option is set to NO, the server will fail the current transaction and discontinue the store operation.

### **ACTIVEDATApools**

Specifies the names of active-data pools where the server simultaneously writes data during a client backup operation. The ACTIVEDATAPOOLS parameter is optional. Spaces between the names of the active-data pools are not permitted.

The combined total number of storage pools specified in the COPYSCTPOOLS and ACTIVEDATAPOOLS parameters cannot exceed three.

When a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of active-data pools from the destination storage pool specified in the copy group. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API.

### Restrictions:

1. This parameter is available only to primary storage pools that use NATIVE or NONBLOCK data format. This parameter is not available for storage pools that use the following data formats:
  - NETAPPDUMP
  - CELERRADUMP
  - NDMPDUMP
2. Simultaneous write to active-data pools is not supported when the operation is using LAN-free data movement. Simultaneous write takes precedence over LAN-free operations, causing the operations to go over the LAN. However, the simultaneous write configuration is honored.
3. Simultaneous write is not supported when a NAS backup operation is writing a TOC file. If the primary storage pool specified in the TOCDESTINATION in the copy group of the management class has active-data pools defined, the active-data pools are ignored and the data is stored into the primary storage pool only.
4. You cannot use the simultaneous write function with Centera storage devices.
5. Data being imported will not be stored in active-data pools. After an import operation, use the COPY ACTIVATEDATA command to store the imported data in an active-data pool.

**Attention:** The function provided by the ACTIVEDATAPOOLS parameter is not intended to replace the COPY ACTIVATEDATA command. If you use the ACTIVEDATAPOOLS parameter, use the COPY ACTIVATEDATA command to ensure that the active-data pools contain all active data of the primary storage pool.

### SHRED

Specifies whether data will be physically overwritten when it is deleted. This parameter is optional. You can specify an integer from 0 to 10.

If you specify a value of 0, the Tivoli Storage Manager server will delete the data from the database. However, the storage used to contain the data will not be overwritten, and the data will still exist in storage until that storage is reused for other data. It might be possible to discover and reconstruct the data after it has been deleted. Changing the value (for example, resetting it to 0) will not affect data that has already been deleted and is currently waiting to be overwritten.

If you specify a value greater than 0, the Tivoli Storage Manager server will delete the data both logically and physically. The server will overwrite the storage used to contain the data the specified number of times. This prevents any attempts to discover and reconstruct the data after it has been deleted.

To ensure that all copies of the data are shredded, specify a SHRED value greater than 0 for the storage pool specified in the NEXTSTGPOOL parameter, and do not specify either the COPYSTGPools or ACTIVEDATAPOOLS. Specifying relatively high values for the overwrite count will generally improve the level of security, but could affect performance adversely.

Overwriting of deleted data is performed asynchronously after the delete operation is complete. Therefore, the space occupied by the deleted data will remain occupied for some period of time and will not be available as free space for new data.

A SHRED value greater than zero cannot be used if the value of the CACHE parameter is YES. If you want to enable shredding for an existing storage pool for which caching is already enabled, you must change the value of the CACHE parameter to NO. Note that existing cached files will remain in storage so that subsequent retrieval requests can be satisfied quickly. If space is needed to store new data, the existing cached files are erased so that the space they occupied can be used for the new data. The existing cached files will not be shredded when they are erased.

**Important:** After an export operation finishes identifying files for export, any changes to the storage pool SHRED value is ignored. An export operation that is suspended retains the original SHRED value throughout the operation. You might want to consider cancelling your export operation if changes to the storage pool SHRED value jeopardize the operation. You can reissue the export command after any needed cleanup.

### **Example: Update a random access storage pool to permit caching**

Update the random access storage pool named BACKUPPOOL to permit caching when the server migrates files to the next storage pool.

```
update stgpool backuppool cache=yes
```

### UPDATE STGPOOL (Update a primary sequential access pool)

Use this command to update a primary sequential access storage pool.

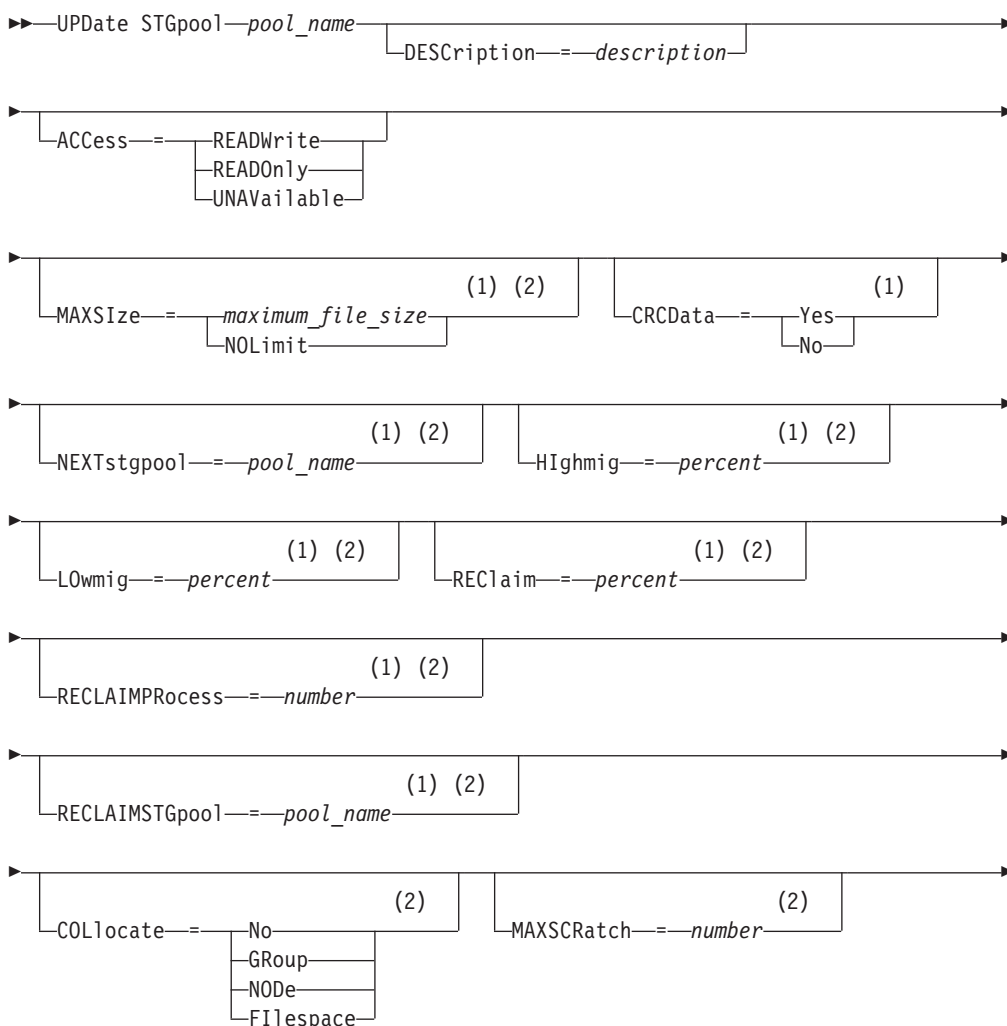
#### Restrictions:

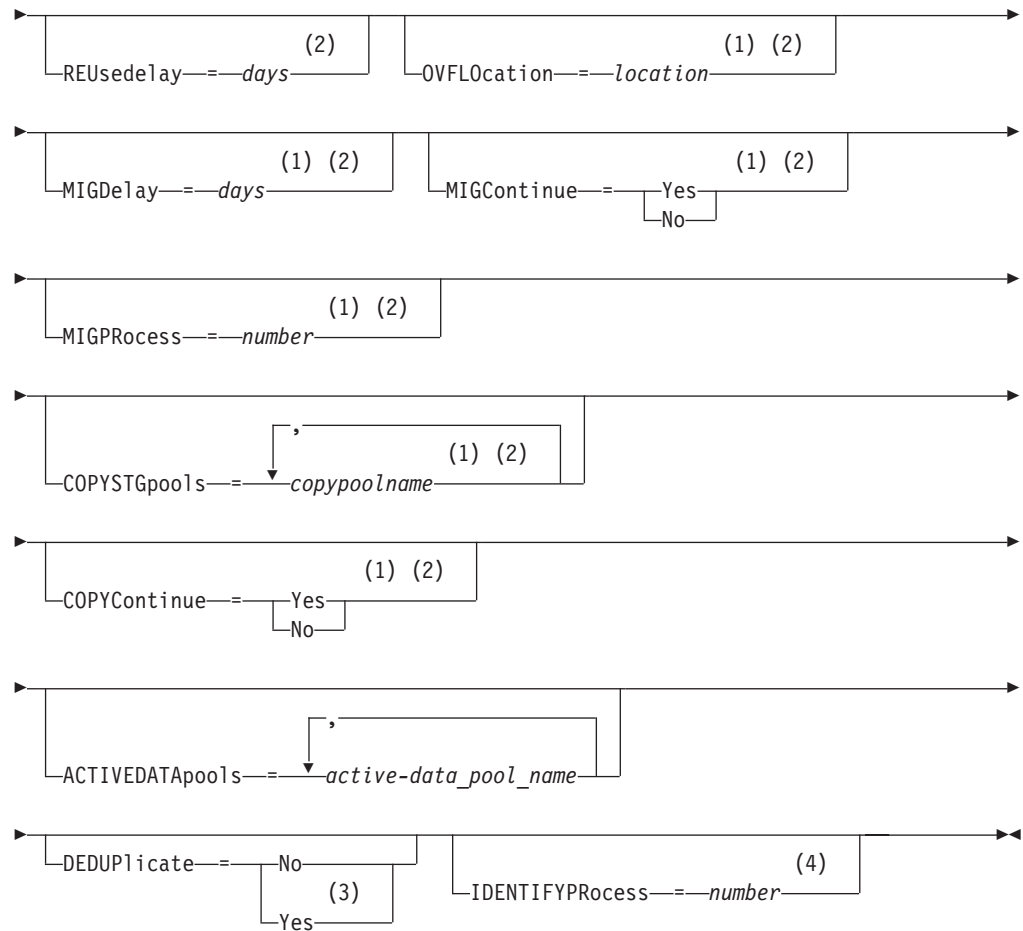
1. You cannot use this command to change the data format for the storage pool.
2. If the value for DATAFORMAT is NETAPPDUMP, CELERRADUMP, or NDMPDUMP, you can modify only the following attributes:
  - DESCRIPTION
  - ACCESS
  - COLLOCATE
  - MAXSCRATCH
  - REUSEDELAY

#### 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





## Notes:

- 1 This parameter is not available for storage pools that use the data formats NETAPPDUMP, CELERRADUMP, or NDMPDUMP.
- 2 This parameter is not available for Centera storage pools.
- 3 This parameter is valid only for storage pools that are defined with a FILE-type device class.
- 4 This parameter is only available if the value of the DEDUPLICATE parameter is YES.

## Parameters

### *pool\_name* (Required)

Specifies the name of the storage pool to be updated.

### DESCription

Specifies a description of the storage pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove an existing description, specify a null string ("").

### ACCess

Specifies how client nodes and server processes (such as migration and reclamation) can access files in the storage pool. This parameter is optional. Possible values are:

### **READWrite**

Specifies that client nodes and server processes can read and write to files stored on volumes in the storage pool.

### **READOnly**

Specifies that client nodes can only read files from the volumes in the storage pool.

Server processes can move files within the volumes in the storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside 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 server processes attempt to write files to the storage pool.

### **UNAvailable**

Specifies that client nodes cannot access files stored on volumes in the storage pool.

Server processes can move files within the volumes in the storage pool and can also move or copy files from this storage pool to another storage pool. However, no new writes are permitted to volumes in the storage pool from volumes outside the 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 server processes attempt to write files to the storage pool.

### **MAXSize**

Specifies the maximum size for a physical file that the server can store in the storage pool. This parameter is optional. Possible values are:

#### **NOLimit**

Specifies that there is no maximum size limit for physical files stored in the storage pool.

#### *maximum\_file\_size*

Limits the maximum physical file size. Specify an integer from 1 to 999999, followed by a scale factor. For example, MAXSIZE=5G specifies that the maximum file size for this storage pool is 5 gigabytes. Scale factors are:

| Scale factor | Meaning |
|--------------|---------|
|--------------|---------|

|   |          |
|---|----------|
| K | kilobyte |
| M | megabyte |
| G | gigabyte |
| T | terabyte |

If a file exceeds the maximum size and no pool is specified as the next storage pool in the hierarchy, the server does not store the file. If a file exceeds the maximum size and a pool is specified as the next storage pool, the server stores the file in the next storage pool that can accept the file size. If you specify the next storage pool parameter, at least one storage pool in your hierarchy should have no limit on the maximum size of a file. By having no limit on the size for at least one pool, you ensure that no matter what its size, the server can store the file.

For logical files that are part of an aggregate, the server considers the size of the aggregate to be the file size. Therefore, the server does not store logical



files that are smaller than the maximum size limit if the files are part of an aggregate that is larger than the maximum size limit.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

#### **CRCData**

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is only valid for NATIVE data format storage pools. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

##### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

##### **No**

Specifies that data is stored without CRC information.

#### **NEXTstgpool**

Specifies a primary storage pool to which files are migrated. You cannot migrate data from a sequential access storage pool to a random access storage pool. This parameter is optional. The next storage pool must be a primary storage pool.

To remove an existing value, specify a null string ("").

If this storage pool does not have a next storage pool, the server cannot migrate files from this storage pool and cannot store files that exceed the maximum size for this storage pool in another storage pool.

When there is insufficient space available in the current storage pool, the NEXTSTGPOOL parameter for sequential access storage pools does not allow data to be stored into the next pool. In this case the server issues a message and the transaction fails.

For next storage pools with a device type of FILE, the server performs a preliminary check to determine whether sufficient space is available. If space is not available, the server skips to the next storage pool in the hierarchy. If space is available, the server attempts to store data in that pool. However, it is possible that the storage operation could fail because, at the time the actual storage operation is attempted, the space is no longer available.

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.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### Highmig

Specifies that the server starts migration when storage pool utilization reaches this percentage. For sequential-access disk (FILE) storage pools, utilization is the ratio of data in a storage pool to the pool's total estimated data capacity, including the capacity of all scratch volumes specified for the pool. For storage pools that use tape or optical media, utilization is the ratio of volumes that contain data to the total number of volumes in the storage pool. The total number of volumes includes the maximum number of scratch volumes. This parameter is optional. You can specify an integer from 0 to 100.

When the storage pool exceeds the high migration threshold, the server can start migration of files by volume to the next storage pool defined for the storage pool. You can set the high migration threshold to 100 to prevent migration for the storage pool.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### LOwmig

Specifies that the server stops migration when storage pool utilization is at or below this percentage. For sequential-access disk (FILE) storage pools, utilization is the ratio of data in a storage pool to the pool's total estimated data capacity, including the capacity of all scratch volumes specified for the pool. For storage pools that use tape or optical media, utilization is the ratio of volumes that contain data to the total number of volumes in the storage pool. The total number of volumes includes the maximum number of scratch volumes. This parameter is optional. You can specify an integer from 0 to 99.

When the storage pool reaches the low migration threshold, the server does not start migration of files from another volume. You can set the low migration threshold to 0 to permit migration to empty the storage pool.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### REClaim

Specifies when the server reclaims a volume, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space on volumes usable again by moving any remaining unexpired 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 to 100.

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 storage pools that use WORM devices, you can lower the value from the default of 100. Lowering the value allows the server to consolidate data onto fewer volumes when needed. Volumes emptied by reclamation can be checked out of the library, freeing slots for new volumes. Because the volumes are write-once, the volumes cannot be reused.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### RECLAIMProcess

Specifies the number of parallel processes to use for reclaiming the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the reclamation and the number of logical and physical drives that can be dedicated to the operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the reclamation.

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that you want to specify four processes for each of the storage pools. The storage pools have the same device class. Assuming that the RECLAIMSTGPOOL parameter is not specified or that the reclaim storage pool has the same device class as the storage pool being reclaimed, each process requires two mount points and, if the device type is not FILE, two drives. (One of the drives is for the input volume, and the other drive is for the output volume.) To run eight reclamation processes simultaneously, you need a total of at least 16 mount points and 16 drives. The device class for the two storage pools must have a mount limit of at least 16.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the reclamation processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### RECLAIMSTGpool

Specifies another primary storage pool as a target for reclaimed data from this storage pool. This parameter is optional. When the server reclaims volumes for the storage pool, unexpired data is moved from the volumes being reclaimed to the storage pool named with this parameter.

To remove an existing value, specify a null string ("").

A reclaim storage pool is most useful for a storage pool that has only one drive in its library. When you specify this parameter, the server moves all data from reclaimed volumes to the reclaim storage pool regardless of the number of drives in the library.

To move data from the reclaim storage pool back to the original storage pool, use the storage pool hierarchy. Specify the original storage pool as the next storage pool for the reclaim storage pool.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### COLlocate

Specifies whether the server attempts to keep data belonging to a single client node, group of client nodes, or client file space 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. For details, see the *Administrator's Guide*.

Possible values are:

#### No

Specifies that collocation is disabled.

#### GRoup

Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible. If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

If you specify COLLOCATE=GROUP but do not define any collocation groups or if you specify COLLOCATE=GROUP but do not add nodes to a collocation group, data is collocated by node. Be sure to consider tape usage when organizing client nodes into collocation groups. For example, if a tape-based storage pool consists of data from grouped and ungrouped nodes and you specify COLLOCATE=GROUP, the server performs the following actions:

- Collocates by group the data for grouped nodes only. Whenever possible, the server collocates data belonging to a group of nodes on a single tape or on as few tapes as possible. Data for a single node can also be spread across several tapes associated with a group.
- Collocates by node the data for ungrouped nodes. Whenever possible, the server stores the data for a single node on a single tape. All available tapes that already have data for the node are used before available space on any other tape is used.

#### NODe

Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible. If the node has multiple file spaces, the server does not attempt to collocate those file spaces. For backward compatibility, COLLOCATE=YES is still accepted by the server to specify collocation at the client node level.

If a storage pool contains data for a node that is a member of a collocation group and you specify COLLOCATE=NODE, the data will be collocated by node not by group.

### Filespace

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

### MAXSCRatch

Specifies the maximum number of scratch volumes that the server can request. This parameter is optional. You can specify an integer from 0 to 100000000. By allowing the server to request scratch volumes, you avoid having to define each volume to be used.

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.

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 the server and returned to the file system.

**Tip:** For server-to-server operations that utilize virtual volumes and that store a small amount of data, consider specifying a value for the MAXSCRATCH parameter that is higher than the value you typically specify for write operations to other types of volumes. After a write operation to a virtual volume, Tivoli Storage Manager marks the volume as FULL, even if the value of the MAXCAPACITY parameter on the device-class definition has not been reached. The Tivoli Storage Manager server does not keep virtual volumes in FILLING status and does not append to them. If the value of the MAXSCRATCH parameter is too low, server-to-server operations can fail.

### REUsedelay

Specifies the number of days that must elapse after all files are 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 to 9999. The value 0 means that a volume can be rewritten or returned to the scratch pool as soon as all files are 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.

### OVFLocation

Specifies the overflow location for the storage pool. The server assigns this location name to a volume that is ejected from the library by the MOVE MEDIA command. This parameter is optional. The location name can be a maximum length of 255 characters. Enclose the location name in quotation marks if the location name contains any blank characters.

To remove an existing value, specify a null string ("").

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP

- NDMPDUMP

### MIGDelay

Specifies the minimum number of days a file must remain in a storage pool before it becomes eligible for migration. All files on a volume must be eligible for migration before the server selects the volume for migration. To calculate a value to compare to the specified MIGDELAY, the server counts the number of days that the file has been in the storage pool.

This parameter is optional. You can specify an integer from 0 to 9999.

If you want the server to count the number of days based only on when a file was stored and not when it was retrieved, use the NORETRIEVEDATE server option.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### MIGContinue

Specifies whether you allow the server to migrate files that do not satisfy the migration delay time. This parameter is optional.

Because you can require that files remain in the storage pool for a minimum number of days, the server may migrate all eligible files to the next storage pool yet not meet the low migration threshold. This parameter allows you to specify whether the server is allowed to continue migration by migrating files that do not satisfy the migration delay time.

Possible values are:

#### Yes

Specifies that, when necessary to meet the low migration threshold, the server continues to migrate files that have not been stored in the storage pool for the number of days specified by the migration delay period.

#### No

Specifies that the server stops migration when no eligible files remain to be migrated, even before reaching the low migration threshold. The server does not migrate files unless the files have been stored in the storage pool for the number of days specified by the migration delay period.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

### MIGPRocess

Specifies the number of parallel processes to use for migrating the files from the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the migration, and the number of logical and physical drives that can be dedicated to the operation. To access a sequential-access volume, Tivoli Storage Manager uses a mount



point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the migration.

For example, suppose you want to simultaneously migrate the files from volumes in two primary sequential storage pools and that you want to specify three processes for each of the storage pools. The storage pools have the same device class. Assuming that the storage pool to which files are being migrated has the same device class as the storage pool from which files are being migrated, each process requires two mount points and, if the device type is not FILE, two drives. (One drive is for the input volume, and the other drive is for the output volume.) To run six migration processes simultaneously, you need a total of at least 12 mount points and 12 drives. The device class for the storage pools must have a mount limit of at least 12.

If the number of migration processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT time, the migration processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of migration processes regardless of the number of volumes that are eligible for migration. For example, if you specify ten migration processes and only six volumes are eligible for migration, the server will start ten processes and four of them will complete without processing a volume.

### **COPYSTGpools**

Specifies the names of copy storage pools where the server simultaneously writes data. Beginning with Tivoli Storage Manager Version 5.3, you can specify a maximum of three copy pool names separated by commas. (In versions earlier than Version 5.3, the maximum number was ten.) Spaces between the names of the copy pools are not permitted. To add or remove one or more copy storage pools, specify the pool name or names that you want to include in the updated list. For example, if the existing copy pool list includes COPY1 and COPY2 and you want to add COPY3, specify `COPYSTGPOOLS=COPY1,COPY2,COPY3`. To remove all existing copy storage pools associated with the primary storage pool, specify a null string ("" ) for the value (for example, `COPYSTGPOOLS=""`).

When specifying a value for the COPYSTGPOOLS parameter, you can also specify a value for the COPYCONTINUE parameter. For additional information, see the COPYCONTINUE parameter.

The combined total number of storage pools specified in the COPYSGTPOOLS and ACTIVEDATAPOOLS parameters cannot exceed three.

The simultaneous write function changed in Tivoli Storage Manager Version 5.3. In Version 5.3 and later, when a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of copy storage pools and the COPYCONTINUE value from the primary storage pool. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to copy storage pools for the following operations:

- Backup and archive operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API
- Migration operations by Tivoli Storage Manager for Space Management clients
- Import operations that involve copying exported file data from external media to a primary storage pool associated with a copy storage pool list

### Restrictions:

1. This parameter is available only to primary storage pools that use NATIVE or NONBLOCK data format. This parameter is not available for storage pools that use the following data formats:
  - NETAPPDUMP
  - CELERRADUMP
  - NDMPDUMP
2. Simultaneous write takes precedence over LAN-free operations, causing the operations go over the LAN. However, the simultaneous write configuration is honored.
3. Simultaneous write is not supported for NAS backup operations. If the primary storage pool specified in the DESTINATION or TOCDESTINATION in the copy group of the management class has copy storage pools defined, the copy storage pools are ignored and the data is stored into the primary storage pool only.
4. You cannot use the simultaneous write function with Centera storage devices.

**Attention:** The function provided by the COPYSTGPools parameter is not intended to replace the BACKUP STGPOOL command. If you use the COPYSTGPools parameter, continue to use the BACKUP STGPOOL command to ensure that the copy storage pools are complete copies of the primary storage pool. There are cases when a copy may not be created. For more information, see the COPYCONTINUE parameter description.

### COPYContinue

Specifies how the server should react to a copy storage pool write failure for any of the copy storage pools listed in the COPYSTGPools parameter. This parameter is optional. The default is YES. When specifying the COPYCONTINUE parameter, either a COPYSTGPools list must already exist or the COPYSTGPools parameter must also be specified.

This parameter has no effect on active-data pools. If a write error occurs while writing to an active-data pool, the server will stop writing to the failing active-data pool for the remainder of the session.

Possible values are:

#### Yes

If the COPYCONTINUE option is set to YES, the server will stop writing to the failing copy pools for the remainder of the session, but continue storing files into the primary pool and any remaining copy pools. The copy storage pool list is active only for the life of the client session and applies to all the primary storage pools in a particular storage pool hierarchy.

For additional information about the COPYCONTINUE parameter, refer to the information about simultaneous write in the *Administrator's Guide*.



**No**

If the COPYCONTINUE option is set to NO, the server will fail the current transaction and discontinue the store operation.

**ACTIVEDATApools**

Specifies the names of active-data pools where the server simultaneously writes data during a client backup operation. The ACTIVEDATAPOOLS parameter is optional. Spaces between the names of the active-data pools are not permitted.

The combined total number of storage pools specified in the COPYSGTPOOLS and ACTIVEDATAPOOLS parameters can not exceed three.

When a data storage operation switches from a primary storage pool to a next storage pool, the next storage pool inherits the list of active-data pools from the destination storage pool specified in the copy group. The primary storage pool is specified by the copy group of the management class that is bound to the data. For details, refer to information about simultaneous write in the *Administrator's Guide*.

The server can perform simultaneous write to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients using the Tivoli Storage Manager API.

**Restrictions:**

1. This parameter is available only to primary storage pools that use NATIVE or NONBLOCK data format. This parameter is not available for storage pools that use the following data formats:
  - NETAPPDUMP
  - CELERRADUMP
  - NDMPDUMP
2. Simultaneous write to active-data pools is not supported when the operation is using LAN-free data movement. Simultaneous write takes precedence over LAN-free operations, causing the operations go over the LAN. However, the simultaneous write configuration is honored.
3. Simultaneous write is not supported when a NAS backup operation is writing a TOC file. If the primary storage pool specified in the TOCDESTINATION in the copy group of the management class has active-data pools defined, the active-data pools are ignored and the data is stored into the primary storage pool only.
4. You cannot use the simultaneous write function with Centera storage devices.
5. Data being imported will not be stored in active-data pools. After an import operation, use the COPY ACTIVEDATA command to store the imported data in an active-data pool.

**Attention:** The function provided by the ACTIVEDATAPOOLS parameter is not intended to replace the COPY ACTIVEDATA command. If you use the ACTIVEDATAPOOLS parameter, use the COPY ACTIVEDATA command to ensure that the active-data pools contain all active data of the primary storage pool.

**DEDuplicate**

Specifies whether the data that is stored in this storage pool will be deduplicated. This parameter is optional and is valid only for storage pools that are defined with a FILE device class.

### IDENTIFYProcess

Specifies the number of parallel processes to use for duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE-type device class. Enter a value from 1 to 20.

When calculating the value for this parameter, consider the workload on the server and the amount of data requiring deduplication. Deduplication identification requires disk I/O and CPU resources, so the more processes you allocate to deduplication, the heavier the workload you place on your system. In addition, consider the number of volumes that require processing. Duplicate-identification processes work on volumes containing data that requires deduplication. If you update a storage pool, specifying that the data in the storage pool is to be deduplicated, all of the volumes in the pool will require processing. For this reason, you might need to define a high number of duplicate-identification processes initially. Over time, however, as existing volumes are processed, only the volumes containing new data will need to be processed. At this time, you can reduce the number of duplicate-identification processes.

**Important:** Duplicate-identification processes can be either active or idle. Processes that are working on files are active. Processes that are waiting for files to work on are idle. Processes remain idle until volumes with data to be deduplicated become available. The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. (For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.) Processes end only when cancelled or when the number of duplicate-identification processes for the storage pool is changed to a value less than that currently specified.

### Example: Update the primary sequential storage pool's mountable scratch volumes

Update the primary sequential storage pool named TAPEPOOL1 to permit as many as 10 scratch volumes to be mounted.

```
update stgpool tapepool1 maxscratch=10
```

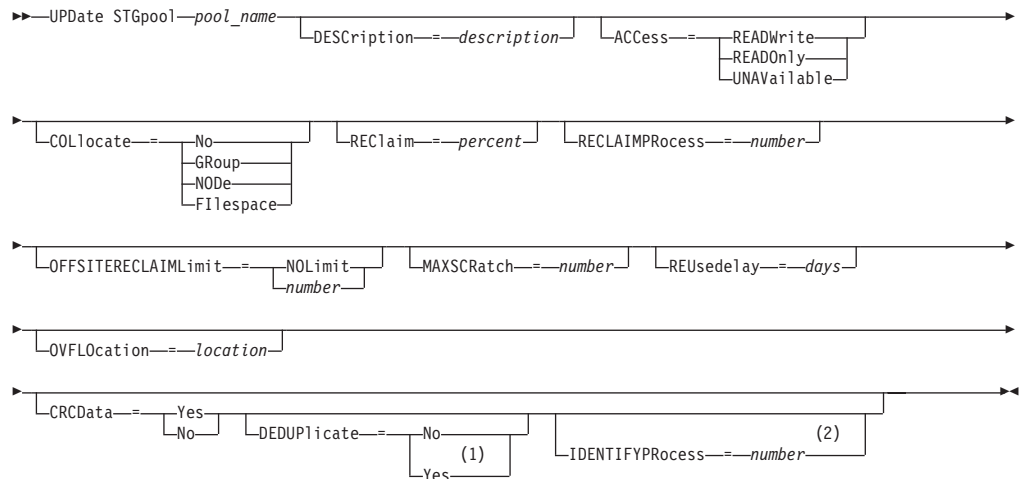
## UPDATE STGPOOL (Update a copy sequential access storage pool)

Use this command to update a copy sequential access storage pool.

### 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



### Notes:

- 1 This parameter is valid only for storage pools that are defined with a FILE-type device class.
- 2 This parameter is only available if the value of the DEDUPLICATE parameter is YES.

### Parameters

#### *pool\_name* (Required)

Specifies the name of the copy storage pool to be updated.

#### DESCRiption

Specifies a description of the copy storage pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove an existing description, specify a null string ("").

#### ACCess

Specifies how client nodes and server processes (such as 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 client nodes can only read files stored on the volumes in the copy storage pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the copy storage pool to restore files to primary storage pools. However, no new writes are permitted to volumes in the copy storage pool from volumes outside the storage pool. A storage pool cannot be backed up to the copy storage pool.

### **UNAVailable**

Specifies that client nodes cannot access files stored on volumes in the copy storage pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the copy storage pool to restore files to primary storage pools. However, no new writes are permitted to volumes in the copy storage pool from volumes outside the storage pool. A storage pool cannot be backed up to the copy storage pool.

### **COLlocate**

Specifies whether the server attempts to keep data belonging to a single client node, group of client nodes, or client file space 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. For details, see the *Administrator's Guide*.

Possible values are:

#### **No**

Specifies that collocation is disabled.

#### **GRoup**

Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible. If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

If you specify COLLOCATE=GROUP but do not define any collocation groups or if you specify COLLOCATE=GROUP but do not add nodes to a collocation group, data is collocated by node. Be sure to consider tape usage when organizing client nodes into collocation groups. For example, if a tape-based storage pool consists of data from grouped and ungrouped nodes and you specify COLLOCATE=GROUP, the server performs the following actions:

- Collocates by group the data for grouped nodes only. Whenever possible, the server collocates data belonging to a group of nodes on a single tape or on as few tapes as possible. Data for a single node can also be spread across several tapes associated with a group.
- Collocates by node the data for ungrouped nodes. Whenever possible, the server stores the data for a single node on a single tape. All available tapes that already have data for the node are used before available space on any other tape is used.

#### **NODe**

Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible. If the node has multiple file spaces, the server does not attempt to collocate those file spaces. For backward compatibility, COLLOCATE=YES is still accepted by the server to specify collocation at the client node level.

If a storage pool contains data for a node that is a member of a collocation group and you specify COLLOCATE=NODE, the data will be collocated by node not by group.

### **Filespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

### **REClaim**

Specifies when the server reclaims a volume, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space on volumes 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 to 100. The value 100 means that reclamation is not performed.

If you change the value from the default of 100, specify a value of 50 percent or greater so that files stored on two volumes can be combined onto a single output volume.

When a copy pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to obtain the active files on the reclaimable volume from a primary or copy storage pool that is onsite. The process then writes these files to an available volume in the original copy storage pool. Effectively, these files are moved back to the onsite location. However, the files could be obtained from the offsite volume after a disaster if a database backup is used that references the files on the offsite volume. Because of the way reclamation works with offsite volumes, use it carefully with copy storage pools.

### **RECLAIMPRocess**

Specifies the number of parallel processes to use for reclaiming the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the reclamation and the number of logical and physical drives that can be dedicated to the operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the reclamation.

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that you want to specify four processes for each of the storage pools. The storage pools have the same device class. Each process requires two mount points and, if the device type is not FILE, two drives. (One of the drives is for the input volume, and the other drive is for the output volume.) To run eight reclamation processes simultaneously, you need a total of at least 16 mount points and 16 drives. The device class for each storage pool must have a mount limit of at least eight.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT

time, the reclamation processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

### **OFFSITERECLAIMLimit**

Specifies the number of offsite volumes to have their space reclaimed during reclamation for this storage pool. This parameter is optional. Possible values are:

#### **NOLimit**

Specifies that you want to have the space reclaimed in all of your offsite volumes.

#### *number*

Specifies the number of offsite volumes to have their space reclaimed. You can specify an integer from 0 to 99999. A value of zero means that none of the offsite volumes will be reclaimed.

**Important:** When determining the value for the OFFSITERECLAIMLIMIT, consider using the statistical information in the message issued at the end of the offsite volume reclamation operation. Alternatively, you can use the following Tivoli Storage Manager SQL select command to obtain the statistical information from the SUMMARY table for the offsite volume reclamation operation:

```
select * from summary where activity='OFFSITE RECLAMATION'
```

The statistical information includes the following items:

- The number of offsite volumes that were processed
- The number of parallel processes that were used
- The total amount of time required for the processing

The order in which offsite volumes are reclaimed is based on the amount of unused space in a volume. (Unused space includes both space that has never been used on the volume and space that has become empty because of file deletion.) Volumes with the largest amount of unused space are reclaimed first.

For example, suppose a copy storage pool contains three volumes: VOL1, VOL2, and VOL3. VOL1 has the largest amount of unused space, and VOL3 has the least amount of unused space. Suppose further that the percentage of unused space in each of the three volumes is greater than the value of the RECLAIM parameter. If you do not specify a value for the OFFSITERECLAIMLIMIT parameter, all three volumes will be reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 will be reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 will be reclaimed.

### **MAXSCRatch**

Specifies the maximum number of scratch volumes that the server can request for this storage pool. This parameter is optional. You can specify an integer from 0 to 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

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.

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 is not deleted from the copy storage pool until the access mode is changed. This allows an administrator to query the server 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 the server and returned to the file system.

**Tip:** For server-to-server operations that utilize virtual volumes and that store a small amount of data, consider specifying a value for the MAXSCRATCH parameter that is higher than the value you typically specify for write operations to other types of volumes. After a write operation to a virtual volume, Tivoli Storage Manager marks the volume as FULL, even if the value of the MAXCAPACITY parameter on the device-class definition has not been reached. The Tivoli Storage Manager server does not keep virtual volumes in FILLING status and does not append to them. If the value of the MAXSCRATCH parameter is too low, server-to-server operations can fail.

#### REUsedelay

Specifies the number of days that must elapse after all files are 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 to 9999. A value of 0 means that a volume can be rewritten or returned to the scratch pool as soon as all files are deleted from the volume.

**Important:** Use this parameter to help ensure that when you restore the database to an earlier level, database references to files in the copy storage pool are still valid. You must set this parameter to a value greater than the number of days you plan to retain the oldest database backup. The number of days specified for this parameter should be the same as the number specified for the SET DRMDBBACKUPEXPIREDDAYS command. For more information, see the *Administrator's Guide*.

#### OVFLocation

Specifies the overflow location for the storage pool. The server assigns this location name to a volume that is ejected from the library by the MOVE MEDIA command. This parameter is optional. The location name can be a maximum length of 255 characters. Enclose the location name in quotation marks if the location name contains any blank characters.

To remove an existing value, specify a null string (").

#### CRCDData

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is only valid for NATIVE data format storage pools. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

##### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts



## UPDATE STGPOOL

performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

**No**

Specifies that data is stored without CRC information.

### **DEDuplicate**

Specifies whether the data that is stored in this storage pool will be deduplicated. This parameter is optional and is valid only for storage pools that are defined with a FILE-type device class.

### **IDENTIFYProcess**

Specifies the number of parallel processes to use for duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value from 1 to 20.

When calculating the value for this parameter, consider the workload on the server and the amount of data requiring deduplication. Deduplication identification requires disk I/O and CPU resources, so the more processes you allocate to deduplication, the heavier the workload you place on your system. In addition, consider the number of volumes that require processing. Duplicate-identification processes work on volumes containing data that requires deduplication. If you update a storage pool, specifying that the data in the storage pool is to be deduplicated, all of the volumes in the pool will require processing. For this reason, you might need to define a high number of duplicate-identification processes initially. Over time, however, as existing volumes are processed, only the volumes containing new data will need to be processed. At this time, you can reduce the number of duplicate-identification processes.

**Important:** Duplicate-identification processes can be either active or idle. Processes that are working on files are active. Processes that are waiting for files to work on are idle. Processes remain idle until volumes with data to be deduplicated become available. The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. (For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.) Processes end only when cancelled or when the number of duplicate-identification processes for the storage pool is changed to a value less than that currently specified.

### **Example: Update a copy storage pool to a 30 day volume reuse and to collocate files by client node**

Update the copy storage pool named TAPEPOOL2 to change the delay for volume reuse to 30 days and to collocate files by client node.

```
update stgpool tapepool2 reusedelay=30 collocate=node
```

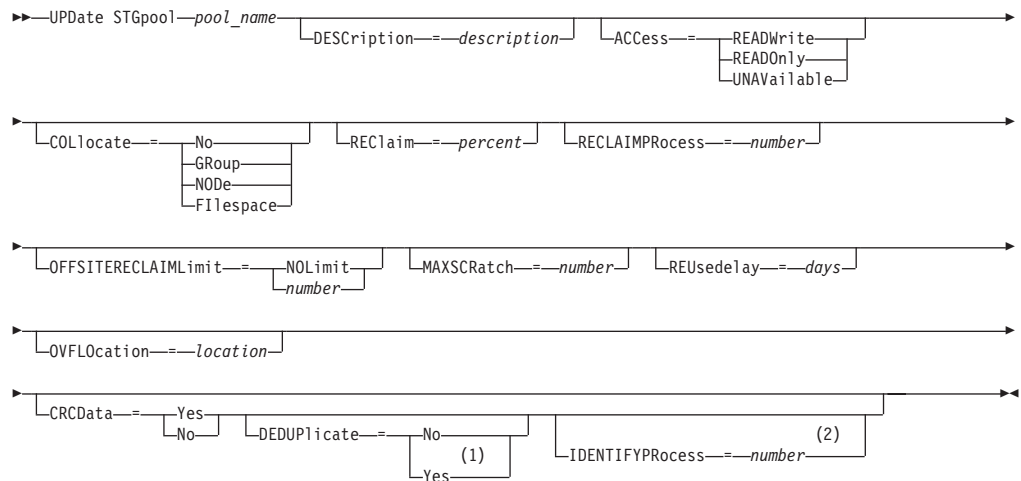


**UPDATE STGPOOL (Update an active-data sequential access)**

Use this command to update an active-data pool.

**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****Notes:**

- 1 This parameter is valid only for storage pools that are defined with a FILE-type device class.
- 2 This parameter is only available if the value of the DEDUPLICATE parameter is YES.

**Parameters*****pool\_name* (Required)**

Specifies the name of the active-data pool to be updated.

**DESCRiption**

Specifies a description of the active-data pool. This parameter is optional. The maximum length of the description is 255 characters. Enclose the description in quotation marks if it contains any blank characters. To remove an existing description, specify a null string ("").

**ACCess**

Specifies how client nodes and server processes (such as reclamation) can access files in the active-data pool. This parameter is optional. Possible values are:

**READWrite**

Specifies that files can be read from and written to the volumes in the active-data pool.

**READOnly**

Specifies that client nodes can only read files stored on the volumes in the active-data pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the active-data pool to restore active versions of backup files to primary storage pools. However, no new writes are permitted to volumes in the active-data pool from volumes outside the storage pool. A storage pool cannot be copied to the active-data pool.

### **UNAVailable**

Specifies that client nodes cannot access files stored on volumes in the active-data pool.

Server processes can move files within the volumes in the storage pool. The server can use files in the active-data pool to restore active versions of backup files to primary storage pools. However, no new writes are permitted to volumes in the active-data pool from volumes outside the storage pool. A storage pool cannot be copied to the active-data pool.

### **COLlocate**

Specifies whether the server attempts to keep data belonging to a single client node, group of client nodes, or client file space 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. For details, see the *Administrator's Guide*.

Possible values are:

#### **No**

Specifies that collocation is disabled.

#### **GRoup**

Specifies that collocation is enabled at the group level for client nodes. The server attempts to put data for nodes that belong to the same collocation group on as few volumes as possible. If the nodes in the collocation group have multiple file spaces, the server does not attempt to collocate those file spaces.

If you specify COLLOCATE=GROUP but do not define any collocation groups or if you specify COLLOCATE=GROUP but do not add nodes to a collocation group, data is collocated by node. Be sure to consider tape usage when organizing client nodes into collocation groups. For example, if a tape-based storage pool consists of data from grouped and ungrouped nodes and you specify COLLOCATE=GROUP, the server performs the following actions:

- Collocates by group the data for grouped nodes only. Whenever possible, the server collocates data belonging to a group of nodes on a single tape or on as few tapes as possible. Data for a single node can also be spread across several tapes associated with a group.
- Collocates by node the data for ungrouped nodes. Whenever possible, the server stores the data for a single node on a single tape. All available tapes that already have data for the node are used before available space on any other tape is used.

#### **NODe**

Specifies that collocation is enabled at the client node level. The server attempts to put data for one node on as few volumes as possible. If the node has multiple file spaces, the server does not attempt to collocate those file spaces. For backward compatibility, COLLOCATE=YES is still accepted by the server to specify collocation at the client node level.

If a storage pool contains data for a node that is a member of a collocation group and you specify COLLOCATE=NODE, the data will be collocated by node not by group.

### **Filespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to put data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to put data for different file spaces on different volumes.

### **REClaim**

Specifies when the server reclaims a volume, based on the percentage of reclaimable space on a volume. Reclamation makes the fragmented space on volumes 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 to 100. The value 100 means that reclamation is not performed.

If you change the value from the default of 60, specify a value of 50 percent or greater so that files stored on two volumes can be combined onto a single output volume.

When an active-data pool volume that is offsite becomes eligible for reclamation, the reclamation process attempts to obtain the active files on the reclaimable volume from a primary or active-data pool that is onsite. The process then writes these files to an available volume in the original active-data pool. Effectively, these files are moved back to the onsite location. However, the files could be obtained from the offsite volume after a disaster if a database backup is used that references the files on the offsite volume. Because of the way reclamation works with offsite volumes, use it carefully with active-data pools.

### **RECLAIMPRocess**

Specifies the number of parallel processes to use for reclaiming the volumes in this storage pool. This parameter is optional. Enter a value from 1 to 999.

When calculating the value for this parameter, consider the number of sequential storage pools that will be involved with the reclamation and the number of logical and physical drives that can be dedicated to the operation. To access a sequential access volume, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive. The number of available mount points and drives depends on other Tivoli Storage Manager and system activity and on the mount limits of the device classes for the sequential access storage pools that are involved in the reclamation.

For example, suppose that you want to reclaim the volumes from two sequential storage pools simultaneously and that you want to specify four processes for each of the storage pools. The storage pools have the same device class. Each process requires two mount points and, if the device type is not FILE, two drives. (One of the drives is for the input volume, and the other drive is for the output volume.) To run eight reclamation processes simultaneously, you need a total of at least 16 mount points and 16 drives. The device class for each storage pool must have a mount limit of at least eight.

If the number of reclamation processes you specify is more than the number of available mount points or drives, the processes that do not obtain mount points or drives will wait for mount points or drives to become available. If mount points or drives do not become available within the MOUNTWAIT

time, the reclamation processes will end. For information about specifying the MOUNTWAIT time, see “DEFINE DEVCLASS (Define a device class)” on page 139.

The Tivoli Storage Manager server will start the specified number of reclamation processes regardless of the number of volumes that are eligible for reclamation. For example, if you specify ten reclamation processes and only six volumes are eligible for reclamation, the server will start ten processes and four of them will complete without processing a volume.

### **OFFSITERECLAIMLimit**

Specifies the number of offsite volumes to have their space reclaimed during reclamation for this storage pool. This parameter is optional. Possible values are:

#### **NOLimit**

Specifies that you want to have the space reclaimed in all of your offsite volumes.

#### *number*

Specifies the number of offsite volumes to have their space reclaimed. You can specify an integer from 0 to 99999. A value of zero means that none of the offsite volumes will be reclaimed.

**Important:** When determining the value for the OFFSITERECLAIMLIMIT, consider using the statistical information in the message issued at the end of the offsite volume reclamation operation. Alternatively, you can use the following Tivoli Storage Manager SQL select command to obtain the statistical information from the SUMMARY table for the offsite volume reclamation operation:

```
select * from summary where activity='OFFSITE RECLAMATION'
```

The statistical information includes the following items:

- The number of offsite volumes that were processed
- The number of parallel processes that were used
- The total amount of time required for the processing

The order in which offsite volumes are reclaimed is based on the amount of unused space in a volume. (Unused space includes both space that has never been used on the volume and space that has become empty because of file deletion.) Volumes with the largest amount of unused space are reclaimed first.

For example, suppose an active-data pool contains three volumes: VOL1, VOL2, and VOL3. VOL1 has the largest amount of unused space, and VOL3 has the least amount of unused space. Suppose further that the percentage of unused space in each of the three volumes is greater than the value of the RECLAIM parameter. If you do not specify a value for the OFFSITERECLAIMLIMIT parameter, all three volumes will be reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 will be reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 will be reclaimed.

### **MAXSCRatch**

Specifies the maximum number of scratch volumes that the server can request for this storage pool. This parameter is optional. You can specify an integer from 0 to 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

The value specified for this parameter is used to estimate the total number of volumes available in the active-data pool and the corresponding estimated capacity for the active-data pool.

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 is not deleted from the active-data pool until the access mode is changed. This allows an administrator to query the server 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 the server and returned to the file system.

**Tip:** For server-to-server operations that utilize virtual volumes and that store a small amount of data, consider specifying a value for the MAXSCRATCH parameter that is higher than the value you typically specify for write operations to other types of volumes. After a write operation to a virtual volume, Tivoli Storage Manager marks the volume as FULL, even if the value of the MAXCAPACITY parameter on the device-class definition has not been reached. The Tivoli Storage Manager server does not keep virtual volumes in FILLING status and does not append to them. If the value of the MAXSCRATCH parameter is too low, server-to-server operations can fail.

#### REUsedelay

Specifies the number of days that must elapse after all files are 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 to 9999. A value of 0 means that a volume can be rewritten or returned to the scratch pool as soon as all files are deleted from the volume.

**Important:** Use this parameter to help ensure that when you restore the database to an earlier level, database references to files in the active-data pool are still valid. You must set this parameter to a value greater than the number of days you plan to retain the oldest database backup. The number of days specified for this parameter should be the same as the number specified for the SET DRMDBBACKUPEXPIREDAYS command. For more information, see the *Administrator's Guide*.

#### OVFLOcation

Specifies the overflow location for the storage pool. The server assigns this location name to a volume that is ejected from the library by the MOVE MEDIA command. This parameter is optional. The location name can be a maximum length of 255 characters. Enclose the location name in quotation marks if the location name contains any blank characters.

To remove an existing value, specify a null string ("").

#### CRCDData

Specifies whether a cyclic redundancy check (CRC) validates storage pool data when audit volume processing occurs on the server. This parameter is only valid for NATIVE data format storage pools. This parameter is optional. The default value is NO. By setting CRCDATA to YES and scheduling an AUDIT VOLUME command you can continually ensure the integrity of data stored in your storage hierarchy. Possible values are:

##### Yes

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts

## UPDATE STGPOOL

performance because additional overhead is required to calculate and compare CRC values between the storage pool and the server.

**No**

Specifies that data is stored without CRC information.

### **DEDuplicate**

Specifies whether the data that is stored in this storage pool will be deduplicated. This parameter is optional and is valid only for storage pools that are defined with a FILE-type device class.

### **IDENTIFYProcess**

Specifies the number of parallel processes to use for duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value from 1 to 20.

When calculating the value for this parameter, consider the workload on the server and the amount of data requiring deduplication. Deduplication identification requires disk I/O and CPU resources, so the more processes you allocate to deduplication, the heavier the workload you place on your system. In addition, consider the number of volumes that require processing. Duplicate-identification processes work on volumes containing data that requires deduplication. If you update a storage pool, specifying that the data in the storage pool is to be deduplicated, all of the volumes in the pool will require processing. For this reason, you might need to define a high number of duplicate-identification processes initially. Over time, however, as existing volumes are processed, only the volumes containing new data will need to be processed. At this time, you can reduce the number of duplicate-identification processes.

**Important:** Duplicate-identification processes can be either active or idle. Processes that are working on files are active. Processes that are waiting for files to work on are idle. Processes remain idle until volumes with data to be deduplicated become available. The output of the QUERY PROCESS command for a duplicate-identification process includes the total number of bytes and files that have been processed since the process first started. (For example, if a duplicate-identification process processes four files, idles, and then processes five more files, then the total number of files processed is nine.) Processes end only when cancelled or when the number of duplicate-identification processes for the storage pool is changed to a value less than that currently specified.

### **Example: Update an active data pool**

Update the active-data pool named TAPEPOOL2 to change the delay for volume reuse to 30 days and to collocate files by client node.

```
update stgpool tapepool3 reusedelay=30 collocate=node
```

## UPDATE VIRTUALFSMAPPING (Update a virtual file space mapping)

Use this command to update an existing virtual file space mapping definition. Refer to the documentation about your NAS device for more information about this command.

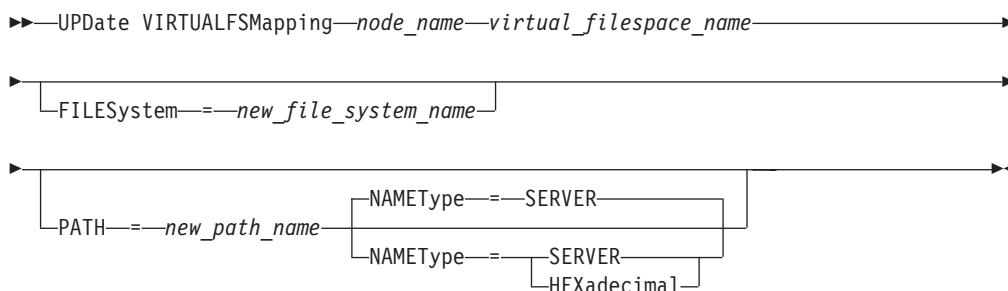
The NAS node must have an associated data mover definition because when the Tivoli Storage Manager server updates a virtual file space mapping, the server attempts to contact the NAS device to validate the virtual file system and file system name.

### Privilege class

To issue this command, you must have one of the following privilege classes:

- System privilege
- Unrestricted policy privilege
- Restricted policy privilege for the domain to which the NAS node is assigned

### Syntax



### Parameters

#### *node\_name* (Required)

Specifies the NAS node on which the file system and path reside. You cannot use wildcard characters or specify a list of names.

#### *virtual\_filespace\_name* (Required)

Specifies the virtual file space mapping to update. You cannot use wildcard characters or specify a list of names.

#### **FileSystem**

Specifies the new name of the file system in which the path is located. The file system name must exist on the specified NAS node. The file system name cannot contain wildcard characters. The file system name should only be modified when the file system name is modified on the NAS device or, for example, the directory is moved to a different file system. This parameter is optional.

#### **PATH**

Specifies the new path from the root of the file system to the directory. The path can only reference a directory. This should only be modified when the path on the NAS device has changed; for example, the directory is moved to a



## UPDATE VIRTUALFSMAPPING

different path. The maximum length of the path is 1024 characters. The path name is case sensitive. This parameter is optional.

### NAMEType

Specifies how the server should interpret the path name specified. Specify this parameter only if you specify a path. This parameter is useful when a path contains characters that are not part of the code page on which the server is running. The default value is SERVER.

Possible values are:

#### SERVER

The code page in which the server is running is used to interpret the path.

#### HEXadecimal

The server interprets the path that you enter as the hexadecimal representation of the path. This option should be used when a path contains characters that cannot be entered. For example, this could occur if the NAS file system is set to a language different from the one in which the server is running.

### Example: Modify the path of a virtual file space mapping

Update the virtual file space mapping named /mikeshomedir for the NAS node NAS1 by modifying the path.

```
update virtualfsmapping nas1 /mikeshomedir path=/new/home/mike
```

### Related commands

Table 384. Commands related to UPDATE VIRTUALFSMAPPING

| Command                 | Description                          |
|-------------------------|--------------------------------------|
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping. |
| DELETE VIRTUALFSMAPPING | Delete a virtual file space mapping. |
| QUERY VIRTUALFSMAPPING  | Query a virtual file space mapping.  |



## UPDATE VOLHISTORY (Update sequential volume history information)

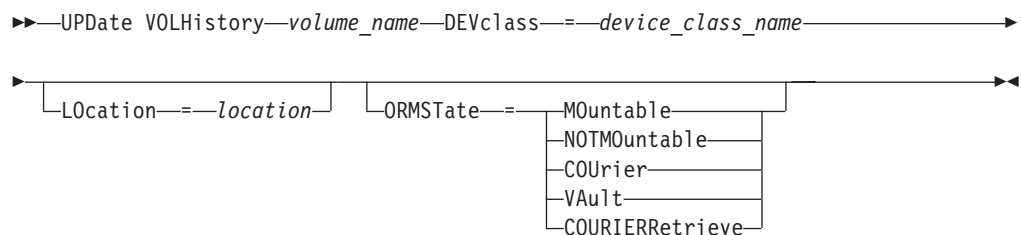
Use this command to update volume history information for a volume produced by a database backup or an export operation. This command does not apply to storage pool volumes.

Use the UPDATE BACKUPSET command to update specified backup set volume information in the volume history file. Do not use this UPDATE VOLHISTORY command to update backup set volume information in the volume history file.

### Privilege class

You must have system privilege or unrestricted storage privilege to issue this command.

### Syntax



### Parameters

#### *volume\_name* (Required)

Specifies the volume name. The volume must have been used for a database backup or an export operation.

#### DEVclass (Required)

Specifies the name of the device class for the volume.

#### LOCATION

Specifies the volume location. This parameter is required if the ORMSTATE parameter is not specified. The maximum text length is 255 characters. Enclose the text in quotation marks if it contains any blank characters.

**Note:** The UPDATE VOLHISTORY command supports updates to the location information and ORMSTATE for snapshot database backup volumes.

#### ORMSTATE

Specifies a change to the state of a database backup volume. This parameter is required if the LOCATION parameter is not specified. This parameter is only supported for systems licensed with Tivoli Disaster Recovery Manager. Possible states are:

##### **MOUNTABLE**

The volume contains valid data and is accessible for on-site processing.

##### **NOTMOUNTABLE**

The volume is on-site, contains valid data, and is not accessible for on-site processing.

##### **COURIER**

The volume is being moved off-site.

## UPDATE VOLHISTORY

### VAult

The volume is off-site, contains valid data, and is not accessible for on-site processing.

### COURIERRetrieve

The volume is being moved on-site.

### Example: Update the location of a volume used for database backup

Update the location of a volume used for database backup, BACKUP1, to show that it has been moved to an off-site location.

```
update volhistory backup1 devclass=tapebkup  
location="700 w. magee rd."
```

### Related commands

Table 385. Commands related to UPDATE VOLHISTORY

| Command           | Description   |
|-------------------|---|
| BACKUP VOLHISTORY | Records volume history information in external files.                                 |
| DELETE VOLHISTORY | Removes sequential volume history information from the volume history file.           |
| MOVE DRMEDIA      | Moves DRM media on-site and off-site.   |
| PREPARE           | Creates a recovery plan file.   |
| QUERY DRMEDIA     | Displays information about disaster recovery volumes.                                 |
| QUERY VOLHISTORY  | Displays sequential volume history information that has been collected by the server. |

UPDATE VOLUME (Change a storage pool volume)

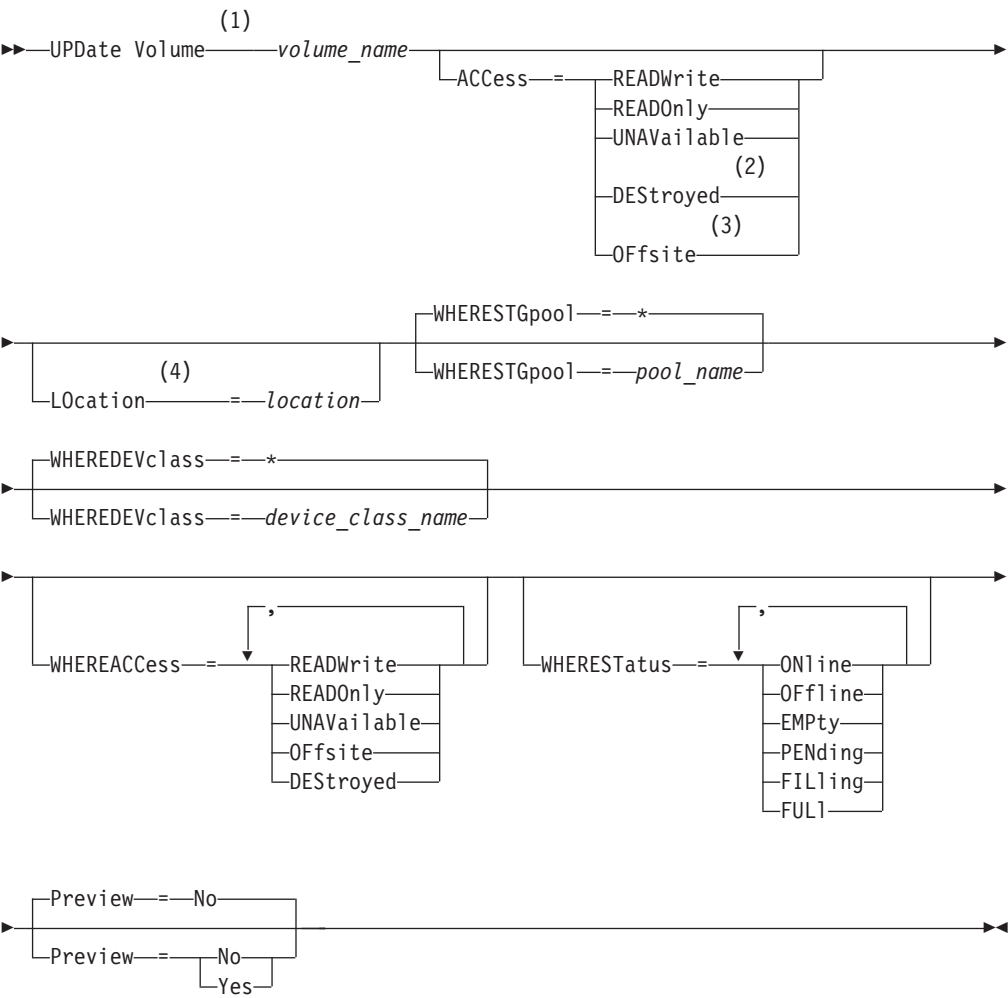
Use this command to change the access mode for one or more volumes in storage pools.

You can correct an error condition associated with a volume by updating the volume to an access mode of READWRITE. You can also use this command to change the location information for one or more volumes in sequential access storage pools.

Privilege class

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

Syntax



Notes:

- 1 You must update at least one attribute (ACCESS or LOCATION).
- 2 This value is valid only for volumes in primary storage pools.
- 3 This value is valid only for volumes in copy storage pools.
- 4 This parameter is valid only for volumes in sequential access storage pools.

### Parameters

#### *volume\_name* (Required)

Specifies the storage pool volume to update. You can use wildcard characters to specify names.

#### **ACcEss**

Specifies how client nodes and server processes (such as migration) can access files in the storage pool volume. This parameter is optional. Possible values are:

##### **READWrite**

Specifies that client nodes and server processes 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 server deletes the volume from the database.

##### **READOnly**

Specifies that client nodes and server processes 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 server deletes the volume from the database.

##### **UNAVailable**

Specifies that neither client nodes nor server processes can access files stored on the volume.

Before making a random access volume unavailable, you must vary the volume offline. After you make a random access volume unavailable, you cannot vary the volume online.

If you make a sequential access volume unavailable, the server 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 server deletes the volume from the database.

##### **DESTroyed**

Specifies that a primary storage pool volume has been permanently damaged. Neither client nodes nor server processes can access files stored on the volume. Use this access mode to indicate an entire volume that needs to be restored by using the RESTORE STGPOOL command. After all files on a destroyed volume have been restored to other volumes, the server automatically deletes the destroyed volume from the database.

Only volumes in primary storage pools can be updated to DESTROYED.

Before updating a random access volume to DESTROYED access, you must vary the volume offline. After you update a random access volume to DESTROYED, you cannot vary the volume online.

If you update a sequential access volume to DESTROYED, the server does not attempt to mount the volume.

If a volume contains no files and you change the access mode to DESTROYED, the server deletes the volume 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 have the access mode of OFFSITE.

Use this mode to help you track volumes that you move to offsite locations. See the *Administrator's Guide* for details.

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 a LOCATION value for a primary storage pool volume, neither the access nor location values are updated because a primary storage pool volume cannot be given an access mode of OFFSITE.

### **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. Enclose the location in quotation marks if it contains any blank characters. To remove a previously defined location, specify the null string ("").

### **WHERESTGpool**

Specifies the name of the storage pool for volumes to be updated. Use this parameter to restrict the update by storage pool. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a storage pool name, volumes belonging to any storage pool are updated.

### **WHEREDEVclass**

Specifies the name of the device class for volumes to be updated. Use this parameter to restrict the update by device class. This parameter is optional. You can use wildcard characters to specify names. If you do not specify a device class name, volumes with any device class are updated.

### **WHEREACcEss**

Specifies the current access mode of volumes to be updated. Use this parameter to restrict the update to volumes that currently have the specified access mode. This parameter is optional. You can specify multiple access modes by separating the modes with commas and no intervening spaces. If you do not specify a value for this parameter, the update is not restricted by the current access mode of a volume. Possible values are:

#### **READWrite**

Update volumes with an access mode of READWRITE.

#### **READOnly**

Update volumes with an access mode of READONLY.

#### **UNAVailable**

Update volumes with an access mode of UNAVAILABLE.

#### **OFFsite**

Update volumes with an access mode of OFFSITE.

#### **DESTroyed**

Update volumes with an access mode of DESTROYED.

### **WHEREStatus**

Specifies the status of volumes to be updated. Use this parameter to restrict the update to volumes that have a specified status. This parameter is optional. You can specify multiple status values by separating the values with commas and no intervening spaces. If you do not specify a value for this parameter, the update is not restricted by volume status. Possible values are:

#### **ONline**

Update volumes with a status of ONLINE.

## UPDATE VOLUME

### OFFline

Update volumes with a status of OFFLINE.

### EMPTy

Update volumes with a status of EMPTY.

### PENding

Update volumes with a status of PENDING. These are volumes from which all files have been deleted, but the time specified by the REUSEDELAY parameter has not elapsed.

### FILLing

Update volumes with a status of FILLING.

### FULl

Update volumes with a status of FULL.

### Preview

Specifies whether you want to preview the update operation without actually updating volumes. This parameter is optional. The default value is NO. Possible values are:

#### No

Specifies that volumes are actually updated.

#### Yes

Specifies that you want only to preview the update operation. This option displays the volumes that will be updated if you actually perform the update operation.

## Example: Make a tape volume unavailable

Update a tape volume named DSMT20 to make it unavailable to client nodes and server processes.

```
update volume dsmt20 access=unavailable
```

## Example: Update the access mode of all offsite volumes in a specific storage pool

Update all empty, offsite volumes in the TAPEPOOL2 storage pool. Set the access mode to READWRITE and delete the location information for the updated volumes.

```
update volume * access=readwrite location="" wherestgpool=tapepool2  
whereaccess=offsite wherestatus=empty
```

## Related commands

Table 386. Commands related to UPDATE VOLUME

| Command       | Description  |
|---------------|--|
| DEFINE VOLUME | Assigns a volume to be used for storage within a specified storage pool. |
| DELETE VOLUME | Deletes a volume from a storage pool.                                    |
| QUERY VOLUME  | Displays information about storage pool volumes.                         |
| VARY          | Specifies whether a disk volume is available to the server for use.      |

---

## VALIDATE commands

Use the VALIDATE command to verify that an object is complete or valid for Tivoli Storage Manager.

- “VALIDATE LANFREE (Validate LAN-Free paths)” on page 1162
- “VALIDATE POLICYSET (Verify a policy set)” on page 1164

## VALIDATE LANFREE (Validate LAN-Free paths)

Use this command to determine which destinations for a given node using a specific storage agent are capable of LAN-Free data movement.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—VALIDate LANfree—*node\_name*—*stgagent\_name*—►►

### Parameters

#### *node\_name* (Required)

The name of the node to evaluate.

#### *stgagent\_name* (Required)

The name of the storage agent to evaluate.

### Example: Validate a current LAN-Free configuration

Validate the current server definitions and configuration for node TIGER to use storage agent AIX\_STA1 for LAN-free data operations.

```
validate lanfree tiger aix_sta1
```

| Node Name | Storage Agent | Operation | Mgmt Class Name | Destination Name | LAN-Free capable? | Explanation   |
|-----------|---------------|-----------|-----------------|------------------|-------------------|---|
| TIGER     | AIX_STA1      | BACKUP    | STANDARD        | OUTPOOL          | NO                | No available online paths. Destination storage pool is configured for simultaneous write. |
| TIGER     | AIX_STA1      | BACKUP    | STANDARD        | PRIMARY          | NO                |   |
| TIGER     | AIX_STA1      | BACKUP    | STANDARD        | SHRPOOL          | YES               | No available online paths. Destination storage pool is configured for simultaneous write. |
| TIGER     | AIX_STA1      | ARCHIVE   | STANDARD        | OUTPOOL          | NO                |   |
| TIGER     | AIX_STA1      | ARCHIVE   | STANDARD        | PRIMARY          | NO                |   |
| TIGER     | AIX_STA1      | ARCHIVE   | STANDARD        | SHRPOOL          | YES               |   |

### Related commands

Table 387. Commands related to VALIDATE LANFREE

| Command         | Description                                       |
|-----------------|---|
| QUERY COPYGROUP | Displays the attributes of a copy group.          |
| QUERY DEVCLASS  | Displays information about device classes.        |
| QUERY DOMAIN    | Displays information about policy domains.        |
| QUERY DRIVE     | Displays information about drives.                |
| QUERY LIBRARY   | Displays information about one or more libraries. |



Table 387. Commands related to VALIDATE LANFREE (continued)

| Command         | Description   |
|-----------------|---|
| QUERY MGMTCLASS | Displays information about management classes.  |
| QUERY NODE      | Displays partial or complete information about one or more clients.                     |
| QUERY PATH      | Displays information about the path from a source to a destination.                     |
| QUERY POLICYSET | Displays information about policy sets.   |
| QUERY SERVER    | Displays information about servers.   |
| QUERY STATUS    | Displays the settings of server parameters, such as those selected by the SET commands. |
| QUERY STGPOOL   | Displays information about storage pools.   |

### VALIDATE POLICYSET (Verify a policy set)

Use this command to verify that a policy set is complete and valid before you activate it. The command examines the management class and copy group definitions in the policy set and reports on conditions that you need to consider before activating the policy set.

The VALIDATE POLICYSET command fails if any of the following conditions exist:

- The policy set has no default management class.
- A copy group within the policy set specifies a copy storage pool as a destination.
- A management class specifies a copy storage pool as the destination for files that were migrated by a Tivoli Storage Manager for Space Management client.
- A TOCDESTINATION parameter is specified, and the storage pool is either a copy pool or has a data format other than NATIVE or NONBLOCK.

The server issues warning messages for the following conditions:

- A copy group specifies a storage pool that does not exist as a destination for backed-up or archived files.  
If you activate a policy set with copy groups that specify nonexistent storage pools, the client backup or archive operations fail.
- A management class specifies a storage pool that does not exist as a destination for files migrated by Tivoli Storage Manager for Space Management clients.
- The policy set does not have one or more management classes that exist in the current ACTIVE policy set.  
If you activate the policy set, backup files bound to the deleted management classes are rebound to the default management class in the new active policy set.
- The policy set does not have one or more copy groups that exist in the current ACTIVE policy set.  
If you activate the policy set, files bound to the management classes with deleted copy groups are no longer archived or backed up.
- The default management class for the policy set does not contain a backup or archive copy group.  
If you activate the policy set with this default management class, clients using the default cannot back up or archive files.
- 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.

If the server has data retention protection enabled, the following conditions must exist:

- All management classes in the policy set to be validated must contain an archive copy group.
- If a management class exists in the active policy set, a management class with the same name must exist in the policy set to be validated.
- If an archive copy group exists in the active policy set, the corresponding copy group in the policy set to be validated must have a RETVER value at least as large as the corresponding values in the active 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 policy set belongs.

## Syntax

►►—VALIDATE POLICYSET—*domain\_name*—*policy\_set\_name*—◄◄

## Parameters

### *domain\_name* (Required)

Specifies the name of the policy domain to which the policy set is assigned.

### *policy\_set\_name* (Required)

Specifies the name of the policy set to be validated.

## Example: Validate a specific policy set

Validate the policy set VACATION located in the EMPLOYEE\_RECORDS policy domain.

```
validate policyset employee_records vacation
```

## Related commands

Table 388. Commands related to VALIDATE POLICYSET

| Command            | Description   |
|--------------------|---|
| ACTIVATE POLICYSET | Validates and activates a policy set.   |
| COPY POLICYSET     | Creates a copy of a policy set.   |
| DEFINE COPYGROUP   | Defines a copy group for backup or archive processing within a specified management class.    |
| DEFINE MGMTCLASS   | Defines a management class.   |
| DELETE POLICYSET   | Deletes a policy set, including its management classes and copy groups, from a policy domain. |
| QUERY POLICYSET    | Displays information about policy sets.   |
| UPDATE COPYGROUP   | Changes one or more attributes of a copy group.   |
| UPDATE POLICYSET   | Changes the description of a policy set.  |

## VARY (Bring a random access volume online or offline)

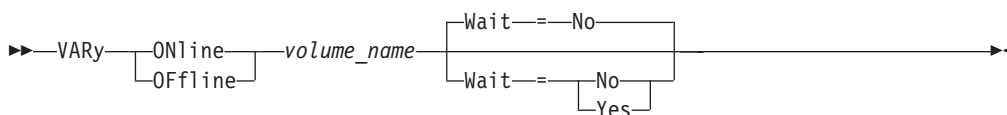
Use this command to make a random access storage pool volume online or offline to the server.

### Privilege class

This command is valid only for volumes on random access devices. For example, use this command during maintenance or corrective action of a random access volume. You cannot vary a random access volume online that is defined as unavailable.

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

### Syntax



### Parameters

#### ONline

Specifies that the server can use the random access volume.

#### OFFline

Specifies that the server cannot use the volume.

#### *volume\_name* (Required)

Specifies the volume identifier. Volume names cannot contain embedded blanks or equal signs.

#### Wait

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is NO. Possible values are:

##### No

Specifies that the server processes this command in the background, while other tasks run. The server displays messages created from the background process either in the activity log or the server console, depending on where messages are logged.

##### Yes

Specifies that the server processes this command in the foreground. Wait for the command to complete before you continue with other tasks. The server displays the output messages to the administrative client when the command completes.

You cannot specify WAIT=YES from the server console.

### Example: Bring volume online

Make volume /adsm/stgv01/1 available to the server for use as a storage pool volume.

```
vary online /adsm/stgv01/1
```

**Related commands***Table 389. Commands related to VARY*

| Command        | Description  |
|----------------|--|
| CANCEL PROCESS | Cancels a background server process.                                     |
| DEFINE VOLUME  | Assigns a volume to be used for storage within a specified storage pool. |
| DELETE VOLUME  | Deletes a volume from a storage pool.                                    |
| QUERY PROCESS  | Displays information about background processes.                         |
| QUERY VOLUME   | Displays information about storage pool volumes.                         |

**VARY**

---

## Chapter 3. Server options

At installation, IBM Tivoli Storage Manager provides a server options file that contains a set of default options to start the server.

The file is:

`dsmserv.opt`

See the *Installation Guide* for your platform to determine where your server options file resides.

Server options let you customize the following:

- Communication
- Server storage
- Client-server
- Date, number, time, and language
- Database and recovery log
- Data transfer
- Message
- Event logging
- Security and licensing

Several other options are available for miscellaneous purposes.

To display the current option settings, enter:

`query option`

---

### Modifying server options

The server reads the server options file at server initialization. When you update a server option by editing the file, you must stop and start the server to activate the updated server options file.

You can change some options dynamically without stopping and starting the server, by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956 for details.

The `dsmserv.opt.smp` file (also provided at installation) contains the format of the options file and all the default settings. You can change any options in the `dsmserv.opt.smp` file. To have the server use the changed options, you must rename the file to `dsmserv.opt`. To activate an option within the server options file, remove the `*>>>` that precedes the option. The server ignores any options preceded by `*>>>`.

---

## Types of server options

Server options let you customize how some functions and processes work.

### Communication options

You can use server options to specify server communication methods and their characteristics.

*Table 390. Communication options*

| Option                | Description   |
|-----------------------|---|
| ADMINONCLIENTPORT     | The port that determines whether administrative sessions can use the port specified in the TCPPORT option   |
| COMMMETHOD            | The server communication method   |
| DNSLOOKUP             | Control of use of Domain Name Services to lookup names of systems contacting the server   |
| NDMPCONTROLPORT       | The internal communications port used for certain Network Data Management Protocol (NDMP) operations  |
| SHMPORT               | The TCP/IP port address of a server when using shared memory  |
| SNMPHEARTBEATINTERVAL | The interval in minutes between queries of the Tivoli Storage Manager server  |
| SNMPMESSAGECATEGORY   | The trap types used when messages are forwarded from the server   |
| SNMPSUBAGENT          | The parameters needed for the Tivoli Storage Manager subagent to communicate with the SNMP daemon   |
| SNMPSUBAGENTHOST      | The location of the Tivoli Storage Manager SNMP subagent  |
| SNMPSUBAGENTPORT      | The port address of the Tivoli Storage Manager SNMP subagent  |
| SSLTCPADMINPORT       | The port address on which the server's TCP/IP communication driver waits for requests for SSL-enabled sessions for the command-line administrative client                                   |
| SSLTCPPOINT           | The SSL-only port number on which the server's TCP/IP communication driver waits for requests for SSL-enabled sessions for the command line backup-archive client and administrative client |
| TCPADMINPORT          | The TCP/IP port number for administrative sessions  |
| TCPBUFSIZE            | The size of the buffer used for TCP/IP send requests  |
| TCPPOINT              | The TCP/IP port number for client sessions  |



Table 390. Communication options (continued)

| Option        | Description                           |
|---------------|---------------------------------------|
| TCPWINDOWSIZE | The client node TCP/IP sliding window |

## Server storage options

You can use server options to configure server storage operations and device operations.

Table 391. Server storage options

| Option              | Description   |
|---------------------|---|
| 3494SHARED          | Enables sharing of a 3494 library with applications other than Tivoli Storage Manager   |
| ACSACCESSID         | The ID for the ACS access control   |
| ACSLOCKDRIVE        | Allows the drives within the ACSLS libraries to be locked   |
| ACSQUICKINIT        | Allows a quick or full initialization of the ACSLS library  |
| ACSTIMEOUTX         | The multiple for the built-in timeout value for the ACSLS API   |
| ASSISTVCRRECOVERY   | Specifies whether the server assists an IBM 3570 or 3590 drive in recovering from a lost or corrupted Vital Cartridge Records (VCR) condition |
| DEVCONFIG           | The name of the file that store backup copies of device configuration information   |
| DRIVEACQUIRERETRY   | The number of times that the server retries the acquisition of a drive in an IBM 349x library that is shared among multiple applications      |
| RECLAIMDELAY        | The number of days that the reclamation of a SnapLock volume is delayed   |
| RECLAIMPERIOD       | The number of days for the reclamation period of a SnapLock volume  |
| RESOURCETIMEOUT     | The length of time that the server waits for a resource before canceling the pending acquisition of a resource                                |
| RETENTIONEXTENSION  | The number of days to extend the retention date of a SnapLock volume  |
| SANDISCOVERY        | Whether the Tivoli Storage Manager SAN discovery function is enabled  |
| SANDISCOVERYTIMEOUT | Amount of time before the SAN discovery process times out   |
| SANREFRESHTIME      | Amount of time before cached SAN discovery information is refreshed   |
| SEARCHMPQUEUE       | The order in which the server satisfies requests in the mount queue   |

## Client-server options

You can use server options to control client-server processing.

Table 392. Client-Server options

| Option        | Description   |
|---------------|---|
| COMMTIMEOUT   | The number of seconds the server waits for a response from a client before timing out the client session            |
| DISABLESCHEDS | Whether administrative and client schedules are disabled during the Tivoli Storage Manager server recovery scenario |
| IDLETIMEOUT   | The number of minutes the server allows a client session to remain idle before timing out the client session        |
| MAXSESSIONS   | The maximum number of simultaneous client sessions with the server  |

Table 392. Client-Server options (continued)

| Option                  | Description  |
|-------------------------|--|
| THROUGHPUTDATATHRESHOLD | The throughput threshold that a client session must reach to prevent being cancelled after the time threshold is reached |
| THROUGHPUTTIMETHRESHOLD | The time threshold for a session after which it may be cancelled for low throughput                                      |
| VERBCHECK               | Whether additional error checking is done for commands sent by the client  |

## Date, number, time, and language options

You can use server options to specify display formats for the dates, times, numbers, and national language.

Table 393. Date, number, time, and language options

| Option   | Description  |
|----------|--|
| LANGUAGE | The national language is used to present client messages |

## Database options

You can use server options to control some aspects of database processing.

Table 394. Database options

| Option             | Description   |
|--------------------|---|
| ACTIVELOGDIR       | The new directory for the location where the active log is stored. Use this option to change the location of the active log.                |
| ACTIVELOGSIZE      | The maximum size of the active log.   |
| ARCHIVELOGDIR      | The directory that the database manager can archive an log file into after all the transactions represented in that log file are completed. |
| ARCHFAILOVERLOGDIR | The directory in which the server tries to store archive log files that cannot be stored in the archive log directory.                      |
| DBMEMPERCENT       | The percentage of system memory that is dedicated to the database.  |
| MIRRORLOGDIR       | The directory for mirroring the active log path.  |

## Data transfer options

You can use server options to control how Tivoli Storage Manager groups and transfers data.

Table 395. Group options

| Option               | Description   |
|----------------------|---|
| MOVEBATCHSIZE        | The number of files that are to be moved and grouped in a batch, within a transaction   |
| MOVESIZETHRESH       | The threshold for the amount of data moved as a batch, within the same server transaction   |
| NDMPPORTRANGE        | The IP address associated with the interface in which the server receives all Network Data Management Protocol (NDMP) backup data |
| NDMPREFDATAINTERFACE | The IP address associated with the interface in which the server receives all Network Data Management Protocol (NDMP) backup data |
| TXNGROUPMAX          | The number of files that are transferred as a group between a client and the server between transaction commit points             |

## Message options

You can use server options to give you more flexibility in the way Tivoli Storage Manager issues messages.

Table 396. Message options

| Option        | Description   |
|---------------|---|
| EXPQUIET      | Whether Tivoli Storage Manager sends detailed informational messages during expiration processing       |
| MESSAGEFORMAT | Whether a message number is displayed in all lines of a multi-line message                              |
| MSGINTERVAL   | The time, in minutes, between messages prompting an operator to mount a tape for Tivoli Storage Manager |

## Event logging options

Options can help you manage event logging receivers.

Table 397. Event logging options

| Option               | Description  |
|----------------------|--|
| EVENTSERVER          | Whether the server should try to contact the event server when the server starts up                |
| FILEEXIT             | A file to which enabled events are routed (binary format)  |
| FILETEXTEXIT         | A file to which enabled events are routed (readable format)  |
| REPORTRETRIEVE       | Record client restore and retrieve operations  |
| TECBEGINEVENTLOGGING | Whether event logging for the Tivoli receiver should begin when the server starts up               |
| TECHOST              | The host name or IP address for the Tivoli event server  |
| TECPORT              | The TCP/IP port address on which the Tivoli event server is listening                              |
| TECUTF8EVENT         | A Tivoli Enterprise Console (TEC) event sent from the Tivoli Storage Manager server in UTF8 format |
| UNIQUETDPTECEVENTS   | Tivoli Data Protection (TDP) events sent to the Tivoli Enterprise Console as unique                |
| UNIQUETECEVENTS      | Events sent to the Tivoli Enterprise Console as unique   |
| USEREXIT             | A user-defined exit that will be given control to manage an event                                  |

## Security options and licensing options

You can use server options to customize server security and license audits.

Table 398. Security and licensing options

| Option            | Description   |
|-------------------|---|
| AUDITSTORAGE      | Specifies that during a license audit operation, the server calculates, by node, the amount of backup, archive, and space management storage in use |
| QUERYAUTH         | The administrative authority level required to issue QUERY or SQL SELECT commands   |
| REQSYSAUTHOUTFILE | Specifies if system authority is required for administrative commands that cause Tivoli Storage Manager to write to an external file                |

## Miscellaneous options

You can use a variety of miscellaneous server options to customize Tivoli Storage Manager.

Table 399. Miscellaneous options

| Option          | Description   |
|-----------------|---|
| AIXASYNCIO      | Specifies whether to enable AIX Asynchronous I/O on AIX   |
| ALIASHALT       | Allows administrators to give the Tivoli Storage Manager HALT command a different name  |
| DISPLAYLFINFO   | Specifies whether accounting records and summary table entries report the storage agent name  |
| EXPINTERVAL     | The interval between automatic inventory expiration processes   |
| NDMPCONTROLPORT | Specifies the port number to be used for internal communications for certain Network Data Management Protocol (NDMP) operations                                       |
| NDMPPORTRANGE   | Specifies the range of port numbers through which Tivoli Storage Manager cycles to obtain a port number for accepting a session from a NAS device for data transfer   |
| NOPREEMPT       | Specifies that no operation can preempt another for access to a volume and that only a database backup operation can preempt another operation for access to a device |
| NORETRIEVEDATE  | Specifies that the server does not update the retrieve date of a file in a disk storage pool when a client restores or retrieves the file                             |
| RESTOREINTERVAL | The length of time that a restartable restore session can be saved in the server database   |
| VOLUMEHISTORY   | The name of the file to be automatically updated whenever server sequential volume history information is changed   |

---

## 3494SHARED

The 3494SHARED option specifies whether an IBM 3494 library can share applications other than Tivoli Storage Manager.

The default is NO, meaning that no application other than Tivoli Storage Manager can share the 3494. When you set this option to YES, for every mount request, Tivoli Storage Manager determines if each drive is in use. After the query completes, Tivoli Storage Manager selects an available drive that is not in use by another application. Enable sharing only if you have more than two drives in your library. If you are currently sharing an IBM 3494 library with other applications, you must specify this option.

### Syntax



### Parameters

#### Yes

Specifies that other applications can share the 3494 library.

#### No

Specifies that no other applications can share the 3494 library.

### Examples

Enable sharing of a 3494 library:

```
3494shared yes
```

---

### ACSACCESSID

The ACSACCESSID option specifies the ID for the ACS access control for an ACSLS library.

#### Syntax

►►—ACSACCESSID—*name*—◄◄

#### Parameters

*name*

Specifies a 1 to 64 character ID. The default ID is your local host name.

#### Examples

acsaccessid region

---

## ACSLOCKDRIVE

The ACSLOCKDRIVE option specifies if the drives within the ACSLS libraries are locked. Drive locking ensures the exclusive use of the drive in the ACSLS library in a shared environment. However, there is some performance gain if libraries are not locked. When other applications do not share the Tivoli Storage Manager drives, drive locking is not required.

### Syntax



### Parameters

#### Yes

Specifies that drives are locked.

#### No

Specifies that drives are not locked.

### Examples

`acslockdrive yes`

---

## ACSQUICKINIT

The ACSQUICKINIT option specifies whether, at server startup, the initialization of the ACSLS library is a quick or full initialization. The default is Yes. A quick initialization avoids the overhead associated with synchronizing the Tivoli Storage Manager server inventory with the ACSLS library inventory (through an audit of the library).

### Syntax

►► ACSQUICKINIT Yes  
No ◄◄

### Parameters

#### Yes

Specifies that a quick initialization of the ACSLS library is performed. When the option is set to Yes, Tivoli Storage Manager bypasses library inventory verification, initializing the library quickly, and making it available to Tivoli Storage Manager sooner than if a full initialization is done.

This option should be set to Yes when it is known that the physical library inventory and the Tivoli Storage Manager library inventory have not changed and an audit is not needed.

#### No

Specifies that a full initialization of the ACSLS library and library inventory is performed. When the option is set to No, Tivoli Storage Manager synchronizes its library volume inventory with what is reported by the ACSLS library manager.

### Examples

```
acsquickinit yes
```



---

## ACSTIMEOUTX

The ACSTIMEOUTX option specifies the multiple for the built-in timeout value for ACSLS APIs. The built-in timeout value for the ENTER, EJECT, and AUDIT ACS API is 1800 seconds; for all other ACSLS APIs it is 600 seconds. For example, if the multiple value specified is 5, the timeout value for audit API becomes 9000 seconds, and all other APIs become 3000 seconds.

### Syntax

►►—ACSTIMEOUTX—*value*—◄◄

### Parameters

*value*

Specifies the multiple for the built-in timeout value for ACSLS API. The range is from 1 to 100. The default is 1.

### Examples

```
acstimeoutx 1
```

---

# ACTIVELOGDIR

The ACTIVELOGDIR option specifies the name of the directory where all active logs are stored.

This option is appended to the options file when the DSMSERV FORMAT command is run. Under normal operating conditions, the option does not need to be changed. See “DSMSERV FORMAT (Format the database and log)” on page 1263 for guidance on this option.

### Syntax

►►—ACTIVELOGDir—*dir\_name*—————►◄

### Parameters

*dir\_name*

Specifies a fully qualified directory name. The directory must already exist, it must be empty, and it must be accessible by the user ID of the database manager. If you change the active log directory, Tivoli Storage Manager moves the existing active logs to the location that is specified by this directory. The maximum number of characters is 175.

### Examples

activelogdir /tsm/activelogdir

---

## ACTIVELOGSIZE

The ACTIVELOGSIZE option sets the total log size.

This option is appended to the options file when the DSMSERV FORMAT command is run. Under normal operating conditions the option does not need to be changed. See “DSMSERV FORMAT (Format the database and log)” on page 1263 for guidance on this option.

### Syntax

►►—ACTIVELOGSize—*megabytes*—◄◄

### Parameters

*megabytes*

Specifies the size of the active log in megabytes. The minimum value is 2048 MB (2 GB); the maximum is 131072 MB (128 GB). If you specify an odd number, the value is rounded up to the next even number. The default is 2048 MB.

### Examples

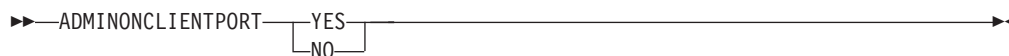
activelogsize 8192

---

### ADMINONCLIENTPORT

The ADMINONCLIENTPORT option defines whether or not the TCPPORT can be used by administrative sessions. The default is YES.

#### Syntax



#### Parameters

##### YES

If the option is set to YES, or if the TCPPORT and TCPADMINPORT are the same value (the default), then administrative sessions can use the TCPPORT.

##### NO

If the option is set to NO, and if the TCPADMINPORT value is different than the TCPPORT value, then administrative sessions cannot use the TCPPORT.

#### Examples

Specify that the TCPPORT can be used by administrative sessions.

```
adminonclientport yes
```

## AIXASYNCIO

The AIXASYNCIO option allows the Tivoli Storage Manager server to use the asynchronous input/output (AIO) support provided on AIX. AIO enhances system performance by allowing individual non-contiguous I/O requests to be gathered together into one request and written to disk in parallel. The default for AIXASYNCIO is NO.

See the AIX documentation for details about AIO before using this option.

AIO is *not* enabled by default on the AIX operating system. To use the AIO feature, you must first enable it on the operating system and restart.

If this server option is specified but AIO has not been enabled on the operating system, an error message is displayed on the server console. The standard I/O method is used instead. If the option is not specified in the server options file, the server behaves as if No was specified.

### Syntax

```

▶▶ AIXASYNCIO { YES | NO }

```

### Parameters

#### YES

Specifies that the Tivoli Storage Manager server will use Asynchronous I/O.

#### NO

Specifies that the Tivoli Storage Manager server will not use Asynchronous I/O.

### Examples

To enable the use of AIXASYNCIO:

```
aixasyncio yes
```

---

## ALIASHALT

The ALIASHALT option allows administrators to give the Tivoli Storage Manager **HALT** command a different name.

The administrative client recognizes an alias for the HALT command when the client is started with the CHECKALIASHALT option specified. See “Administrative client options” on page 5 for details.

### Syntax

►►—ALIASHALT—*newname*—————►◄

### Parameters

*newname*

Specifies the alias of the HALT command for shutting down the Tivoli Storage Manager server. Minimum length of *newname* is 1; maximum length is 16.

### Examples

alias halt tsmhalt

---

## ARCHFAILOVERLOGDIR

The ARCHFAILOVERLOGDIR option specifies the directory which the server uses to store archive log files that cannot be stored in the archive log directory.

This option is appended to the options file when the DSMSEV FORMAT command is run. Typically the directory does not need to be changed.

### Syntax

►►—ARCHFailoverlogdir—*dir\_name*—◄◄

### Parameters

*dir\_name*

Specifies a fully qualified directory name. The maximum number of characters is 175.

### Examples

```
archfailoverlogdir /tsm/archfailoverlog
```

### ARCHIVELOGDIR

The ARCHIVELOGDIR option specifies a directory that the database manager can archive a log file into after all the transactions represented in that log file are completed.

This option is appended to the options file when the DSMSEV FORMAT command is run.

#### Syntax

►►—ARCHIVELOGDir—*dir\_name*—►►

#### Parameters

*dir\_name*

Specifies a fully qualified directory name. The maximum number of characters is 175.

#### Examples

archivelogdir /tsm/archlog



---

## ASSISTVCRRECOVERY

The ASSISTVCRRECOVERY option specifies whether Tivoli Storage Manager assists an IBM 3570 or 3590 drive in recovering from a lost or corrupted Vital Cartridge Records (VCR) condition. If you specify YES (the default) and if Tivoli Storage Manager detects an error during the mount processing, it locates to the end-of-data during the dismount processing to allow the drives to restore the VCR. During the tape operation, there may be some small effect on performance because the drive cannot perform a fast locate with a lost or corrupted VCR. However, there is no loss of data.

### Syntax

▶▶ ASSISTVCRRECOVERY ☐ Yes ☐ No ▶▶

### Parameters

#### Yes

Specifies server assistance in recovery.

#### No

Specifies no server assistance in recovery.

### Examples

Turn off recovery assistance:

```
assistvcrrecovery no
```

---

## AUDITSTORAGE

As part of a license audit operation, the server calculates, by node, the amount of server storage used for backup, archive, and space-managed files. 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 AUDITSTORAGE option to specify that storage is not to be calculated as part of a license audit.

**Note:** This option was previously called NOAUDITSTORAGE.

### Syntax

►►—AUDITStorage—

|     |
|-----|
| Yes |
| No  |

—►►

### Parameters

#### Yes

Specifies that storage is to be calculated as part of a license audit. The default is Yes.

#### No

Specifies that storage is not to be calculated as part of a license audit.

### Examples

auditstorage    yes

## COMMMETHOD

The COMMMETHOD option specifies a communication method to be used by the server. You can specify multiple COMMMETHOD options in the server options file.

### Syntax



### Parameters

You can choose one of the following communication methods:

#### NONE

Specifies that no communication method is used. This option does not allow users to connect to the server and is useful for experimenting with policy commands.

#### SHARED MEM

Specifies the shared memory communication method option. This method uses the same area of memory to send data between several applications at the same time. Both the server and the backup-archive client need to be configured to support the shared memory communication method, and they must be installed on the same machine.

#### SNMP

Specifies the SNMP communication method option.

#### TCPIP

Specifies the TCP/IP communication method option. This is the default. When TCPIP is specified, TCP/IP Version 4 is used exclusively.

#### V6TCPIP

Specifies the TCP/IP communication method option. If TCP/IP Version 4 and Version 6 are both configured, Tivoli Storage Manager uses both protocols simultaneously. If both COMMMETHOD TCPIP and COMMMETHOD V6TCPIP are specified, V6TCPIP overrides the specification of TCPIP. A valid domain name server (DNS) environment must be present to use either TCP/IP Version 4 or TCP/IP Version 6 if this option is specified.

### Examples

```

commmethod tcpip
commmethod v6tcpip
  
```

---

## COMMTIMEOUT

The COMMTIMEOUT option specifies how long the server waits for an expected client message during an operation that causes a database update. If the length of time exceeds this time-out, the server ends the session with the client. You may want to increase the time-out value to prevent clients from timing out. Clients may time out if there is a heavy network load in your environment or they are backing up large files.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—COMMTIMEOUT—60—seconds—◄◄

### Parameters

*seconds*

Specifies the maximum number of seconds that a server waits for a client response. The default value is 60. The minimum value is 1.

### Examples

commtimeout 60

---

## DBMEMPERCENT

Use this option to specify the percentage of the virtual address space that is dedicated to the database manager processes.

If applications other than Tivoli Storage Manager server are running on the system, ensure that the value allows adequate memory for the other applications.

### Syntax

►► DBMEMPERCENT *percent*  
AUTO ◀◀

### Parameters

*percent*

Set a value from 10 to 99.

**AUTO**

The database manager sets the percentage automatically to a value of approximately 70 to 80 percent of system RAM. The default value is AUTO.

### Examples

dbmempercent 50

---

## DEDUPREQUIRESBACKUP

The DEDUPREQUIRESBACKUP option specifies whether volumes in primary sequential-access storage pools that are set up for deduplication can be reclaimed and whether duplicate data can be discarded before the storage pools are backed up.

If the value of this option is YES (the default), you must back up data to copy storage pools that are not set up for deduplication. Use the BACKUP STGPPOOL command to back up data to copy storage pools.

Be aware that reclamation of a volume in a storage pool that is set up for deduplication might not occur when the volume first becomes eligible. The server makes additional checks to ensure that data from a storage pool that is set up for deduplication has been backed up to a copy storage pool. These checks require more than one BACKUP STGPPOOL instance before the server reclaims a volume. After the server verifies that the data was backed up, the volume is reclaimed.

You can change this option dynamically using the SETOPT command.

**Attention:** To minimize the possibility of data loss, do not change the default setting for this server option. Specify a value of NO only if you do not have any copy storage pools and are not performing storage pool backups.

### Syntax

►► DEDUPREQUIRESBACKUP ☐ Yes ☒ No ◀◀

### Parameters

#### Yes

Specifies that the storage pool must be backed up before volumes can be reclaimed and before duplicate data can be discarded. This is the default.

#### No

Specifies that volumes in primary sequential-access storage pools that are set up for deduplication can be reclaimed and duplicate data can be discarded if the storage pools are not backed up.

### Examples

Specify that primary sequential-access storage pools that are set up for deduplication do not have to be backed up.

```
deduprequiresbackup no
```

## DEVCONFIG

The DEVCNFIG option specifies the name of a file in which you want Tivoli Storage Manager to store a backup copy of device configuration information.

Tivoli Storage Manager 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
- Library definitions created by using the DEFINE LIBRARY command
- Library inventory information for the LIBTYPE=SCSI automated libraries
- Path definitions created by using the DEFINE PATH command
- Server definitions created with the DEFINE SERVER command
- Server name created with the SET SERVERNAME command
- Sever password created with the SET SERVERPASSWORD command

### Note:

- Only path definitions with SRCTYPE=SERVER are backed up to the device configuration file. Paths of SRCTYPE=DATAMOVER are *not* written to the file.
- Library volume location information is stored as comments (/...\*/) in the device configuration file whenever CHECKIN LIBVOLUME, CHECKOUT LIBVOLUME, and AUDIT LIBRARY commands are issued for SCSI libraries.

**Attention:** To restore the database after a disaster, you must have a copy of the current device configuration file. The device configuration file cannot be recreated.

You can include one or more DEVCNFIG options in the server options file. When you use multiple DEVCNFIG options, Tivoli Storage Manager 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 to store a backup copy of device configuration information.

## Examples

devconfig devices.sav

## DISABLESCHEDS

The DISABLESCHEDS option specifies whether administrative and client schedules are disabled during Tivoli Storage Manager server recovery.

### Syntax

```

▶▶—DISABLEScheds—Yes  
No—————▶▶

```

### Parameters

#### Yes

Specifies that administrative and client schedules are disabled.

#### No

Specifies that administrative and client schedules are enabled.

### Examples

```
disablescheds no
```

## DISKSTGPPOOLMEMSIZE

The DISKSTGPPOOLMEMSIZE option specifies the size of the memory that the server needs to efficiently manage disk storage pools.

The more memory available, the less disk storage pool metadata must be retrieved from the database server. Performance might be improved during operations that store data into or delete data from disk storage pools.

### Syntax

```

▶▶— DISKSTGPPOOLMem size—megabytes—————▶▶

```

### Parameters

#### *megabytes*

Specifies, in megabytes, the size of the memory available to manage disk storage pools. Each megabyte can manage 32 gigabytes of disk storage. This option should be large enough to accommodate the maximum amount of data expected to be stored in or deleted from disk storage pools per second. For example, if a maximum of 96 gigabytes of data per second is expected to be stored in or deleted from disk storage pools, a size of 3 is recommended. If this option is not specified, it defaults to 80, which can manage 2560 gigabytes of disk storage. For 32-bit servers, it defaults to 20 which can manage 640 gigabytes of disk storage.

### Example:

```
diskstgpoolmemsize 25
```



## DISPLAYLFINFO

The DISPLAYLFINFO option specifies how the accounting records and summary table entries report the node name.

When this option is enabled, the accounting records and summary table entries report *node\_name(storage\_agent\_name)* for the node name. If the option is not enabled, the accounting records and summary table entries simply report *node\_name* for the node name. The default is No.

### Syntax



### Parameters

#### Yes

Specifies that the accounting records and summary table entries will report the storage agent name.

#### No

Specifies that the accounting records and summary table entries will not report the storage agent name. This is the default.

### Examples

```
displaylfinfo yes
```

The result shows the following accounting record with the storage agent name displayed (STA53):

```
5,0,ADSM,07/13/2004,15:35:14,COLIND-TUC(STA53),,WinNT,1,Tcp/Ip,1,0,0,0,
0,223,4063,0,0,222,7,8,3,1,4,0,0,0,0,3,0
```

The corresponding summary table also displays the storage agent name:

```

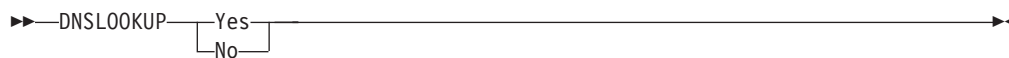
START_TIME: 2004-07-13 15:35:07.000000
END_TIME: 2004-07-13 15:35:14.000000
ACTIVITY: BACKUP
NUMBER: 8
ENTITY: COLIND-TUC(STA53)
COMMMETH: Tcp/Ip
ADDRESS: colind-tuc:2229
SCHEDULE_NAME:
EXAMINED: 0
AFFECTED: 223
FAILED: 0
BYTES: 4160875
IDLE: 8
MEDIAM: 1
PROCESSES: 1
SUCCESSFUL: YES
VOLUME_NAME:
DRIVE_NAME:
LIBRARY_NAME:
LAST_USE:
COMM_WAIT: 3
NUM_OFFSITE_VOLS:
  
```

---

## DNSLOOKUP

The DNSLOOKUP option specifies whether the server uses system API calls to determine the domain name server (DNS) names of systems that contact the server.

### Syntax



### Parameters

#### Yes

Specifies that the server obtains the DNS names of contacting systems. Yes is the default.

#### No

Specifies that the server does not obtain the DNS names of contacting systems.

### Examples

```
dnslookup yes
```

## DRIVEACQUIRERETRY

The DRIVEACQUIRERETRY option lets you specify how many times the server retries the acquisition of a drive in an IBM 349x library. If the library is shared among multiple applications, its drives may appear to be available to the server (through the use of a background polling process) when they are not.

This option is only valid if you specified 3494SHARED YES in the dsmserv.opt file. If you specified DRIVEACQUIRERETRY NEVER, you need to monitor how long jobs have been waiting for drives and how long the server has been polling the drives. You may also need to check the status of these drives in the other Tivoli Storage Manager servers. There may be cartridges stuck in the drives, and the other Tivoli Storage Manager servers may have marked the drives as *offline*. If this is the case, you need to mark the drives *offline* in the Tivoli Storage Manager server that is polling the drives. If necessary, also cancel any waiting jobs.

### Syntax



### Parameters

#### Forever

The acquisition of a drive is retried until one is successfully acquired. This is the default.

#### Never

The server does not retry the acquisition of a drive and fails the operation.

#### *number\_of\_retries*

Specifies the maximum number of times, from 1 to 9999, that the server retries the acquisition of a drive.

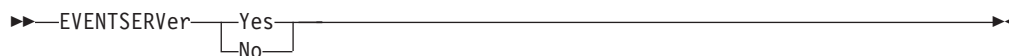
### Examples

Specify that the server should attempt no more than 10 times to acquire the drive:

```
driveacquireretry 10
```

The `EVENTSERVER` option specifies whether at startup the server should try to contact the event server.

## Syntax



## Parameters

**Yes**

Specifies that, at startup, the server tries to contact the event server. Contact occurs only if a `DEFINE EVENTSERVER` command has already been issued. This is the default.

**No**

Specifies that, at startup, the server does not try to contact the event server.

## Examples

```
eventserver yes
```

## EXPINTERVAL

The EXPINTERVAL option specifies the interval, in hours, between automatic inventory expiration processes by Tivoli Storage Manager. 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. If expiration is not run periodically, storage pool space is not reclaimed from expired client files, and the server requires more storage space than required by policy.

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.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—EXPINterval—<sup>24</sup>hours—►►

### Parameters

*hours*

Specifies the time, in hours, between automatic inventory expiration processes. You can specify from 0 to 336 (14 days). A value of 0 means that expiration must be started with the EXPIRE INVENTORY command. The default is 24.

### Examples

```
expinterval 5
```

---

## EXPQUIET

The EXPQUIET option specifies whether Tivoli Storage Manager sends detailed messages during expiration processing.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—EXPQUIET—☐—No—☐—Yes—►►

### Parameters

#### No

Specifies that the server sends detailed messages. This is the default.

#### Yes

Specifies that the server sends only minimal messages. These messages are sent only for files that have expired based on the copy group in the default management class or retention grace period for the domain.

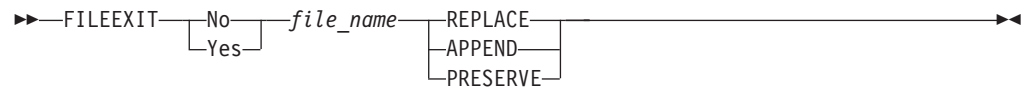
### Examples

```
expquiet no
```

## FILEEXIT

The FILEEXIT option specifies a file to which enabled events are routed. Each logged event is a record in the file.

### Syntax



### Parameters

#### Yes

Specifies that event logging to the file exit receiver begins automatically at server startup.

#### No

Specifies that event logging to the file exit receiver does not begin automatically at server startup. When this parameter has been specified, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.

#### *file\_name*

Specifies the name of the file in which the events are stored.

#### REPLACE

Specifies that if the file already exists, it will be overwritten.

#### APPEND

Specifies that if the file already exists, data is appended to it.

#### PRESERVE

Specifies that if the file already exists, it will not be overwritten.

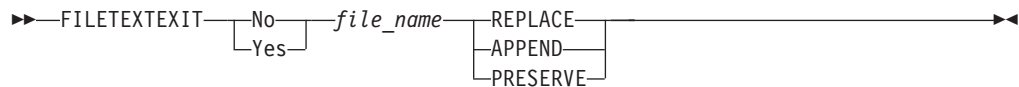
### Examples

```
fileexit yes /tsm/server/data replace
```

## FILETEXTEXIT

The FILETEXTEXIT option specifies a file to which enabled events are routed. Each logged event is a fixed-size, readable line.

### Syntax



### Parameters

#### Yes

Specifies that event logging to the file exit receiver begins automatically at server startup.

#### No

Specifies that event logging to the file exit receiver does not begin automatically at server startup. When this parameter has been specified, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.

#### *file\_name*

Specifies the name of the file in which the events are stored.

#### REPLACE

Specifies that if the file already exists, it will be overwritten.

#### APPEND

Specifies that if the file already exists, data will be appended to it.

#### PRESERVE

Specifies that if the file already exists, it will not be overwritten.

### Examples

```
filetextexit yes /tsm/server/data replace
```



---

## IDLETIMEOUT

The IDLETIMEOUT option specifies the amount of time, in minutes, that a client session can be idle before the server cancels the session. You may want to increase the time-out value to prevent clients from timing out if there is a heavy network load in your environment. Note, however, that a large number of idle sessions could prevent other users from connecting to the server.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—IDLETimeout—<sup>15</sup>  
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
```

## LANGUAGE

The LANGUAGE option controls the initialization of locales. A locale includes the language and the date, time, and number formats to be used for the console and server.

If your client and server are running different languages, the messages generated may not be understandable when messages are issued from the client to the server or if the server sends output to the client.

If initialization of the locale fails, the server defaults to American English.

### Syntax



### Parameters

#### en\_US

Specifies that American English will be used as the default language for the server.

#### locale

Specifies the name of the locale supported by the server. See the following tables for information on supported locales by operating system.

**Note:** Tivoli Storage Manager runs in any locale, but defaults to American English. For the locales listed, language support is available. Refer to the *Installation Guide* for information on installing language support.

Table 400. Server languages for AIX

| Language                      | LANGUAGE option value |
|-------------------------------|-----------------------|
| Chinese, Simplified           | zh_CN                 |
| Chinese, Simplified (UTF-8)   | ZH_CN                 |
| Chinese, Traditional (Big5)   | Zh_TW                 |
| Chinese, Traditional (UTF-8)  | ZH_TW                 |
| Chinese, Traditional (euc_tw) | zh_TW                 |
| English                       | en_US                 |
| English (UTF-8)               | EN_US                 |
| French                        | fr_FR                 |
| French (UTF-8)                | FR_FR                 |
| German                        | de_DE                 |
| German (UTF-8)                | DE_DE                 |
| Italian                       | it_IT                 |
| Italian (UTF-8)               | IT_IT                 |
| Japanese, EUC                 | ja_JP                 |
| Japanese, PC                  | Ja_JP                 |
| Japanese, UTF8                | JA_JP                 |

Table 400. Server languages for AIX (continued)

| Language   | LANGUAGE option value |
|--|-----------------------|
| Korean   | ko_KR                 |
| Korean (UTF-8)   | KO_KR                 |
| Portuguese, Brazilian  | pt_BR                 |
| Portuguese, Brazilian (UTF-8)  | PT_BR                 |
| Russian  | ru_RU                 |
| Russian (UTF-8)  | RU_RU                 |
| Spanish  | es_ES                 |
| Spanish (UTF-8)  | ES_ES                 |
| <b>Table note:</b> The system must have en_US environment support installed. |                       |

## Examples

```
lang ja_JP
```

---

## MAXSESSIONS

The MAXSESSIONS option specifies the maximum number of simultaneous client sessions that can connect with the server.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►► MAXSessions 25  
number\_of\_sessions ◄◄

### Parameters

*number\_of\_sessions*

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 storage size or communication resources.

### Examples

maxsessions 25

---

## MESSAGEFORMAT

The MESSAGEFORMAT option specifies whether a message number is displayed in all lines of a multi-line message.

### Syntax

►►—MESSageformat—*number*—◄◄

### 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.

- 1 The message number for a message is displayed only in the first line of the message. This is the default.
- 2 The message number for a message is displayed in all lines of a message.

### Examples

messageformat 2

---

## MIRRORLOGDIR

The MIRRORLOGDIR option specifies the directory for mirroring the active log path.

All changes made to the active log directory are also written to this mirror directory. This option is appended to the options file when the DSMSEVER FORMAT command is run. Typically, the directory does not need to be changed.

### Syntax

►►—MIRRORlogdir—*dir\_name*—————►◄

### Parameters

*dir\_name*

Specifies a fully qualified directory name for the active log mirror. The maximum number of characters is 175.

### Examples

mirrorlogdir /tsm/mirrorlog

---

## MOVEBATCHSIZE

The MOVEBATCHSIZE option specifies the number of client files that are to be moved and grouped together in a batch, within the same server transaction. This data movement results from storage pool backups and restores, migration, reclamation, and MOVE DATA operations. This option works with the MOVESIZETHRESH option.

### Syntax

►► MOVEBatchsize <sup>1000</sup>  
*number\_of\_files* ◄◄

### Parameters

*number\_of\_files*

Specifies a number of files between 1 and 1000. The default is 1000.

### Examples

movebatchsize 100

---

## MOVESIZETHRESH

The MOVESIZETHRESH option 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

►►—MOVESizethresh—<sup>2048</sup>  
                                  *megabytes*—►►

### Parameters

*megabytes*

Specifies the number of megabytes as an integer from 1 to 2048. The default value is 2048. This option works in conjunction with the MOVEBATCHSIZE option.

### Examples

movesizethresh 500



---

## MSGINTERVAL

The MSGINTERVAL option specifies the time, in minutes, between messages prompting an operator to mount a tape for the server.

### Syntax

►►—MSGINTERval—<sup>1</sup>  
minutes—◄◄

### Parameters

*minutes*

Specifies the time interval at which the operator is prompted by the server to mount a tape. The default value is 1 minute. The minimum value is 1 minute.

### Examples

msginterval 2

---

## NDMPCONTROLPORT

The NDMPCONTROLPORT option specifies the port number to be used for internal communications for certain Network Data Management Protocol (NDMP) operations. The Tivoli Storage Manager server does not function as a general purpose NDMP tape server.

### Syntax

►►—NDMPControlport—<sup>10000</sup>  
                                  *port\_number*—◄◄

### Parameters

*port\_number*

The port number to be used for internal communications for certain NDMP operations. The port number must be from 1024 to 32767. The default is 10000.

### Examples

ndmpcontrolport 9999

## NDMPPORTRANGE

The NDMPPORTRANGE option specifies the range of port numbers through which Tivoli Storage Manager cycles to obtain a port number for accepting a session from a network-attached storage (NAS) device for data transfer. The default is 0,0 which means that Tivoli Storage Manager lets the operating system provide a port (ephemeral port).

If all ports specified are in use when a NAS device attempts to connect to the server, the operation fails. If a single port number is chosen (no comma and no port number for the high value), the default for the high port number is the low port number plus 100.

When Network Data Management Protocol (NDMP) data is directed to a Tivoli Storage Manager native pool, communication can be initiated from either the NDMP systems or the Tivoli Storage Manager server. If a firewall separates the server and NAS devices, it may be necessary to specify port numbers in firewall rules to allow traffic to pass to and from the NAS devices. NAS devices communicate to the Tivoli Storage Manager server the port numbers that they will use when contacting the server. The port numbers of the server are controlled with the NDMPPortrange options. Port number control for NAS devices is specific to vendors. Consult your vendor documentation.

### Syntax

```
NDMPPortrange port_number_low [,port_number_high]
```

### Parameters

*port\_number\_low*

The low port number from which Tivoli Storage Manager starts to cycle when needing a port number for accepting session from a NAS device for data transfer. The minimum port number value is 1024.

*port\_number\_high*

The high port number to which Tivoli Storage Manager can cycle when needing a port number for accepting session from a NAS device for data transfer. The maximum port number value is 32767. The high port number must be the same or larger than the low port number.

### Examples

Specify that Tivoli Storage Manager can cycle from port numbers 1024 - 2024.

```
ndmpportrange 1024,2024
```

---

## NDMPPREFDATAINTERFACE

This option specifies the IP address associated with the interface in which you want the server to receive all Network Data Management Protocol (NDMP) backup data.

This option affects all subsequent NDMP filer-to-server operations, but does not affect NDMP control connections, which use the system's default network interface. The value for this option is a host name or IPV4 address that is associated with one of the active network interfaces of the system on which the Tivoli Storage Manager server is running. This interface must be IPV4 enabled.

You can update this server option without stopping and restarting the server by using the SETOPT command. For more information, see the Administrator's Guide.

### Syntax

►►—NDMPPREFDATAINTERFACE—*ip\_address*—◄◄

### Parameters

*ip\_address*

Specify an address in either dotted decimal or host name format. If you specify a dotted decimal address, it is not verified with a domain name server. If the address is not correct, it can cause failures when the server attempts to open a socket at the start of an NDMP filer-to-server backup.

Host name format addresses are verified with a domain name server. There is no default value. If a value is not set, all NDMP operations will use the Tivoli Storage Manager server's network interface for receiving backup data during NDMP filer-to-server backup operations. To clear the option value, specify the SETOPT command with a null value, "".

### Examples:

```
ndmpprefdatainterface net1.tucson.ibm.com
```

```
ndmpprefdatainterface 9.11.152.89
```

---

## NOPREEMPT

The server allows certain operations to preempt other operations for access to volumes and devices. You can specify the NOPREEMPT option to disable preemption. When preemption is disabled, no operation can preempt another for access to a volume, and only a database backup operation can preempt another operation for access to a device.

For example, a client data restore operation preempts a client data backup for use of a specific device or access to a specific volume.

See the *Administrator's Guide* for details.

### Syntax

►►—NOPREEMPT—◄◄

### Parameters

None

### Examples

Disable preemption among server operations:

```
nopreempt
```

---

## **NORETRIEVEDATE**

The **NORETRIEVEDATE** option specifies that the server does not update the retrieve date of a file in a disk storage pool when a client restores or retrieves the file. This option and the **MIGDELAY** storage pool parameter control when the server migrates files.

If you do not specify **NORETRIEVEDATE**, the server migrates files after they have been in the storage pool for the number of days specified by the **MIGDELAY** parameter. The number of days is counted from the day that the file was stored in the storage pool or retrieved by a client, whichever is more recent. If you specify **NORETRIEVEDATE**, the server does not update the retrieve date of a file, and the number of days is counted from the day the file entered the disk storage pool.

If you specify this option and caching is enabled for a disk storage pool, reclamation of cached space is affected. When space is needed in a disk storage pool that contains cached files, the server gets the space by selectively erasing cached copies. Files that have the oldest retrieve dates and occupy the largest amount of space are selected for removal. When you specify **NORETRIEVEDATE**, the server does not update the retrieve date when a file is retrieved. This may cause cached copies to be removed even though they have recently been retrieved by a client.

### **Syntax**

►►—**NORETRIEVEDATE**—————◄◄

### **Parameters**

None.

### **Examples**

Specify that the retrieve dates of files in disk storage pools are not updated when clients restore and retrieve the files:

```
noretrieve date
```

## NUMOPENVOLSALLOWED

The NUMOPENVOLSALLOWED option specifies the number of input FILE volumes in a deduplicated storage pool that can be open at one time.

Input volumes contain data to be read during client-restore operations and server processes, such as reclamation and migration. Use this option to improve performance by reducing the frequency with which volumes are opened and closed.

Each session within a client operation or server process can have as many open FILE volumes as specified by this option. A session is initiated by a client operation or by a server process. Multiple sessions can be started within each.

During a client restore operation, volumes can remain open for the duration of a client restore operation and as long a client session is active. During a no-query restore operation, the volumes remain open until the no-query restore completes. At that time, all volumes are closed and released. However, for a classic restore operation started in interactive mode, the volumes might remain open at the end of the restore operation. The volumes are closed and released when the next classic restore operation is requested.

Set this value in the server options file or use the SETOPT command.

**Tip:** This option can significantly increase the number of volumes and mount points in use at any one time. To optimize performance, follow these steps:

- To set NUMOPENVOLSALLOWED, select a beginning value (the default is recommended). Monitor client sessions and server processes. Note the highest number of volumes open for a single session or process. Increase the setting of NUMOPENVOLSALLOWED if the highest number of open volumes is equal to the value specified by NUMOPENVOLSALLOWED.
- To prevent sessions or processes from having to wait for a mount point, increase the value of the MOUNTLIMIT parameter in the device-class definition. Set the value of the MOUNTLIMIT parameter high enough to allow all client sessions and server processes using deduplicated storage pools to open the number of volume specified by the NUMOPENVOLSALLOWED option. For client sessions, check the destination in the copy group definition to determine how many nodes are storing data in the deduplicated storage pool. For server processes, check the number of processes allowed for each process for the storage pool.
- A situation might occur in which a node backs up and restores or archives and retrieves concurrently to and from a deduplicated storage pool. All the mount points required for these operations increase the total number of mount points required by the node.

As a result, the node might not be able to start additional backup sessions if it already has more mount points open than what the MAXNUMMP parameter in the client-node definition allows. This can occur even though the MOUNTLIMIT for the device class was not exceeded.

To prevent backup and retrieve operations from failing, set the value of the MAXNUMMP parameter in the client-node definition to a value at least as high as the NUMOPENVOLSALLOWED option. Increase this value if you notice that the node is failing backup or retrieve operations because the MAXNUMMP value is being exceeded.

## NUMOPENVOLSALLOWED

### Syntax

►►—NUMOPENVOLSa1lowed—*number\_of\_open\_volumes*—◄◄

### Parameters

*number\_of\_open\_volumes*

Specifies the number of input FILE volumes in a deduplicated storage pool that can be open at one time. The default is 10. To disable this function and return to legacy behavior, set this option to 1. Doing so allows only one input volume to be open at a time.

### Examples

Specify that up to 5 volumes in a deduplicated storage pool can be open at one time.

```
numopenvolsallowed 5
```



# QUERYAUTH

The QUERYAUTH option specifies the administrative authority level required to issue QUERY or SQL SELECT commands. By default any administrator can issue QUERY and SELECT commands. You can use this option to restrict the use of these commands.

## Syntax



## Parameters

### NOne

Any administrator can issue QUERY or SELECT commands without requiring any administrative authority.

### SYStem

Administrators must have SYSTEM authority to issue QUERY or SELECT commands.

### POLicy

Administrators must have POLICY authority over one or more policy domains or SYSTEM authority to issue QUERY or SELECT commands.

### STorage

Administrators must have STORAGE authority over one or more storage pools or SYSTEM authority to issue QUERY or SELECT commands.

### OPerator

Administrators must have OPERATOR or SYSTEM authority to issue QUERY or SELECT commands.

## Examples

To restrict the use of QUERY and SELECT commands to administrators with system or storage authority, enter:

```
queryauth storage
```

---

## RECLAIMDELAY

This option delays the reclamation of a SnapLock volume, allowing remaining data to expire so that there is no need to reclaim the volume.

### Syntax

►► RECLAIMDELAY <sup>4</sup>  
number\_of\_days ◄◄

### Parameters

*number\_of\_days*

Specifies the number of days to delay the reclamation of a SnapLock volume.

Before a SnapLock volume is reclaimed, the Tivoli Storage Manager server allows the specified number of days to pass, so that any files remaining on the volume have a chance to expire. The default reclaim delay period is 4 days and can be set anywhere from 1 to 120 days.

### Examples

Specify that the number of days to delay reclamation is 30 days:

```
reclaimdelay 30
```

---

## RECLAIMPERIOD

This option allows you to set the number of days for the reclamation period of a SnapLock volume.

### Syntax

►► RECLAIMPERIOD number\_of\_days ◀◀

### Parameters

*number\_of\_days*

Specifies the number of days that are allowed for the reclamation period of a SnapLock volume.

After the retention of a SnapLock volume has expired, the Tivoli Storage Manager server will reclaim the volume within the specified number of days if there is still data remaining on the volume. The default reclaim period is 30 days and can be set anywhere from 7 to 365 days.

The reclamation period does not begin until the RECLAIMDELAY period has expired.

### Examples

Specify that the reclaim period is 45 days:

```
reclaimperiod 45
```

---

## REPORTRETRIEVE

The REPORTRETRIEVE option reports on restore or retrieve operations that are performed by client nodes or administrators. The default is NO.

### Syntax

►►—REPORTRETRIEVE—YES  
                    └─NO┐

### Parameters

#### YES

Specifies that messages will be issued to the server console and stored in the activity log whenever files are restored or retrieved from the Tivoli Storage Manager server. The messages will specify the name of the objects being restored or retrieved and identify the client node or administrator performing the operation.

#### NO

Specifies that messages will not be issued.

### Examples

Specify that messages will be issued and stored in the activity log whenever files are restored or retrieved from the IBM Tivoli Storage Manager server:

```
reportretrieve yes
```

The following message is issued for an administrator client session:

```
ANR0411I Session 8 for administrator COLIND-TUC logged in as node  
COLIND-TUC restored or retrieved Backup object: node COLIND-TUC,  
filesystem \\colind-tuc\c$, object\CODE\TESTDATA\ XXX.OUT
```

---

## REQSYSAUTHOUTFILE

The REQSYSAUTHOUTFILE option specifies if system authority is required for administrative commands that cause Tivoli Storage Manager to write to an external file.

This option applies to the following commands:

- BACKUP DEVCONFIG with the FILENAMES parameter
- BACKUP VOLHISTORY with the FILENAMES parameter
- DEFINE BACKUPSET
- DELETE BACKUPSET
- MOVE DRMEDIA with the CMD parameter
- MOVE MEDIA with the CMD parameter
- QUERY DRMEDIA with the CMD parameter
- QUERY MEDIA with the CMD parameter
- QUERY SCRIPT with the OUTPUTFILE parameter

### Syntax

►►—REQSYSauthoutfile—☐Yes—  
                                  ☐No                                  ◄◄

### Parameters

#### Yes

System authority is required for administrative commands that cause Tivoli Storage Manager to write to an external file.

#### No

System authority is not required for administrative commands that cause Tivoli Storage Manager to write to an external file. That is, there is no change to the authority level that is required to issue the command.

### Examples

```
reqsysauthoutfile no
```

## RESOURCETIMEOUT

The RESOURCETIMEOUT option specifies how long the server waits for a resource before canceling the pending acquisition of a resource. When a timeout occurs the request for the resource will be canceled.

**Note:** When managing a set of shared library resources, such as servers designated as library managers and clients, consider setting this option at the same time limit for all participants in the shared configuration. In any case of error recovery, Tivoli Storage Manager will always defer to the longest time limit.

### Syntax

```

▶▶—RESOURCetimeout—60minutes—▶▶
  
```

### Parameters

*minutes*

Specifies the maximum number of minutes that the server waits for a resource. The default value is 60 minutes. The minimum value is 1 minute.

### Examples

Specify that the server will wait 15 minutes for a server resource:

```
resourcetimeout 15
```

## RESTOREINTERVAL

The RESTOREINTERVAL option specifies how long a restartable restore session can be saved in the server database. As long as the restore session is saved in the database, it can be restarted from the point at which it stopped.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—RESTOREINTERVAL—<sup>1440</sup>  
minutes—►►

### Parameters

*minutes*

Specifies how long, in minutes, that a restartable restore session can be in the database before it can be expired. The minimum value is 0. The maximum is 10080 (one week). The default is 1440 minutes (24 hours). If the value is set to 0 and the restore is interrupted or fails, the restore is still put in the restartable state. However, it is immediately eligible to be expired.

### Examples

```
restoreinterval 1440
```

## RETENTIONEXTENSION

The RETENTIONEXTENSION option specifies the number of days to extend the retention date of a SnapLock volume. This option allows the server to extend the retention date of a SnapLock volume in order to avoid excessive reclamation.

### Syntax

►►—RETENTIONEXTENSION—*number\_of\_days*—◄◄

### Parameters

#### *number\_of\_days*

Specifies the number of days to extend the retention date of a SnapLock volume. The minimum value is 30 days; the maximum value is 9999 days; the default is 365.

If you specify a value of 0 (zero) for the **RETVER** parameter of an archive copy group, the actual value that is used for **RETVER** is the value of the option RETENTIONEXTENSION, if one of the following conditions is also true:

- The destination storage pool for the archive copy group is a SnapLock storage pool.
- The storage pool that is the target for a storage pool migration or of a MOVE DATA or MOVE NODEDATA command is a SnapLock storage pool.

If a SnapLock volume is the target volume for data from another SnapLock volume and if the remaining retention of the data on the volume is less than the value specified, then the retention date is set using the value specified. Otherwise, the remaining retention of the data is used to set the retention of the volume.

If a SnapLock volume has entered the reclamation period but the percentage of reclaimable space of the volume has not exceeded the reclamation threshold of the storage pool or the value specified on the **THRESHOLD** parameter of a RECLAIM STGPOOL command, then the retention date of the SnapLock volume is extended by the amount specified in the RETENTIONEXTENSION option.

### Examples

Specify that the retention date is extended by 60 days:

```
retentionextension 60
```



## SANDISCOVERY

The SANDISCOVERY option specifies whether the Tivoli Storage Manager SAN discovery function is enabled.

When set to ON, the server will perform SAN discovery in the following instances:

- During server initialization
- When the device path has been changed
- When the QUERY SAN command is issued

Using SAN discovery, the server can automatically correct the device's special file name if it has been changed.

To use SAN discovery, all devices on the SAN must have a unique worldwide name (WWN). Some virtual tape libraries use the same WWN for all of their drives. SAN discovery will fail for these libraries. If this happens, set the SANDISCOVERY option to OFF and manually define or update the drive paths. If the virtual tape library provides no WWNs, SAN discovery will attempt to use serial numbers to identify the drives.

The Tivoli Storage Manager server does not require persistent binding with the SAN discovery function enabled. To display a list of devices seen by the server, you can issue the QUERY SAN command.

### Syntax



### Parameters

#### ON

With this option specified, the server will perform SAN discovery during server initialization, when the device path has been changed, or when the QUERY SAN command is issued.

#### OFF

With this option specified, the server will not perform SAN discovery during the server initialization, when the device path has been changed, or when the QUERY SAN command is issued. This is the default.

#### PASSIVE

With this option specified, the server will not perform SAN discovery during server initialization, when the device path has been changed, or when the QUERY SAN command is issued. This conceals the message produced when you disable SAN discovery and does not take device paths off-line.

### Examples

sandiscovery on

---

## SANDISCOVERYTIMEOUT

The SANDISCOVERYTIMEOUT option specifies the amount of time allowed for host bus adapters to respond when they are queried by the SAN discovery process. Once the time specified for the SANDISCOVERYTIMEOUT is reached, the process times out.

### Syntax

►►—SANDISCOVERYTIMEOUT—*value*—◄◄

### Parameters

*value*

Specifies the amount of time to elapse before the SAN discovery process times out. The range is from 15 to 300 seconds. The default is 15 seconds.

### Examples

sandiscoverytimeout 45

## SANREFRESHTIME

The SANREFRESHTIME option specifies the amount of time that elapses before the cached SAN discovery information is refreshed. The SANREFRESHTIME option has a default value of 0, which means there is no SAN discovery cache. Each time the server performs a SAN discovery operation, the information is obtained directly from the host bus adapter (HBA).

**Note:** The QUERY SAN server command always receives SAN information at the time the command is issued and ignores any value specified for SANREFRESHTIME.

### Syntax



### Parameters

*time*

The length of time, in seconds, before the cached SAN discovery information is refreshed. The default value is 0 and specifies that SAN discovery information is not cached. If a value other than 0 is specified, for example, 100 seconds, then the SAN discovery information is refreshed 100 seconds after the prior SAN discovery operation.

### Examples

Refresh SAN discovery information after 100 seconds.

```
sanrefreshtime 100
```

Turn off the caching of SAN discovery information.

```
sanrefreshtime 0
```

---

## SEARCHMPQUEUE

The SEARCHMPQUEUE option specifies the order in which the server satisfies requests in the mount queue. If the option is specified, the server first tries to satisfy requests for volumes that are already mounted. These requests may be satisfied before other requests, even if the others have been waiting longer for the mount point. If this option is not specified, the server satisfies requests in the order in which they are received.

### Syntax

►►—SEARCHMPQUEUE—◄◄

### Parameters

None

### Examples

Specify that the server tries to first satisfy a request for a volume that is already mounted:

```
searchmpqueue
```

---

## SHMPORT

The SHMPORT option specifies the TCP/IP port address of a server when using shared memory. All shared memory communications start with a TCP/IP connection.

### Syntax

►►—SHMPort—*port\_number*—◄◄

### Parameters

*port\_number*

Specifies the port number. You can specify a value from 1024 to 32767. The default value is 1510.

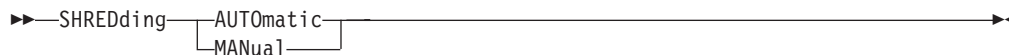
### Examples

```
shmport 1580
```

## SHREDDING

The SHREDDING option specifies whether shredding of deleted sensitive data is performed automatically or manually. Shredding applies only to data in storage pools that have been explicitly configured to support shredding.

### Syntax



### Parameters

#### AUTOMATIC

Specifies that shredding occurs automatically as sensitive data is deleted. Use this option to shred sensitive data as soon as possible after it is deleted. If the SHREDDING option is not specified, this is the default behavior. If there is an I/O error during automatic shredding, an error is reported, and shredding of the current object halts. If the I/O error cannot be corrected, you might need to run shredding manually and use the IOERROR keyword.

#### MANUAL

Specifies that shredding occurs manually, only when the SHRED DATA command is invoked. Use this option to control when shredding takes place, in order to ensure that it does not interfere with other server activities.

**Tip:** If you specify manual shredding, run the SHRED DATA command regularly, at least as often as you perform other routine server-maintenance tasks (for example, expiration, reclamation, and so on). Doing so can prevent performance degradation of certain server processes (in particular, migration). For best results, run SHRED DATA after any operation (for example, expiration and migration) that deletes files from a shred pool.

### Examples

Specify that Tivoli Storage Manager automatically shreds data in a storage pool configured for shredding after that data is deleted:

```
shredding automatic
```

---

## SNMPHEARTBEATINTERVAL

The SNMPHEARTBEATINTERVAL option specifies the interval in minutes between queries of the Tivoli Storage Manager server.

### Syntax

►►—SNMPHEARTBEATINTERVAL—<sup>5</sup>  
*minutes*—◄◄

### Parameters

*minutes*

Specifies the heartbeat interval in minutes. Valid values are from 0 to 1440 (one day). The default is 5 minutes.

### Examples

`snmpheartbeatinterval 20`

---

# SNMPMESSAGECATEGORY

The SNMPMESSAGECATEGORY option specifies the trap types used when messages are forwarded from the server, through the Simple Network Management Protocol (SNMP) subagent, to the SNMP manager.

## Syntax



## Parameters

### SEVERITY

Specifies that there are four trap types based on message severity level:

- 1 Severe
- 2 Error
- 3 Warning
- 4 Information

This is the default.

### INDIVIDUAL

Specifies that a separate trap type is used for each message. The numeric part of the message identifier indicates the trap type.

## Examples

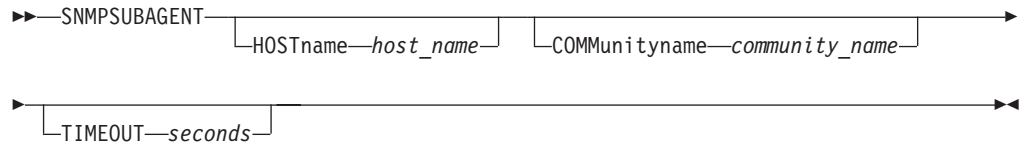
```
snmpmessagecategory individual
```



## SNMPSUBAGENT

The SNMPSUBAGENT option specifies the parameters needed for the Tivoli Storage Manager subagent to communicate with the Simple Network Management Protocol (SNMP) daemon. This option is only to configure the SNMP subagent for communicating with the SNMP agent; it is ignored by the server.

### Syntax



### Parameters

#### HOSTname *host\_name*

Specifies the TCP/IP name or number of the host running the SNMP agent that the Tivoli Storage Manager SNMP subagent connects to. This parameter is optional. The default name is *localhost*.

#### COMMunityname *community\_name*

Specifies the configured community name on the system running the SNMP agent. This parameter is optional. The default name is *public*.

#### TIMEOUT *seconds*

Specifies the time, in seconds, in which a request must be received. This parameter is optional. The default value is 600.

### Examples

```
snmpsubagent hostname jimbo communityname public timeout 2600
```

---

### SNMPSUBAGENTHOST

The SNMPSUBAGENTHOST option specifies the location of the Tivoli Storage Manager Simple Network Management Protocol (SNMP) subagent. The default for this option is 127.0.0.1.

#### Syntax

►►—SNMPSUBAGENTHOST—*host\_name*—◄◄

#### Parameters

*host\_name*

Specifies the TCP/IP host name or number on which the Tivoli Storage Manager SNMP subagent is located. The subagent and server must be on the same node.

#### Examples

snmpsubagenthost 9.116.23.450

---

## SNMPSUBAGENTPORT

The SNMPSUBAGENTPORT option specifies the port number of the Tivoli Storage Manager Simple Network Management Protocol (SNMP) subagent.

### Syntax

►►—SNMPSUBAGENTPORT—*port\_number*—◄◄

### Parameters

*port\_number*

Specifies the port number of the Tivoli Storage Manager SNMP subagent. Valid values are 1000 - 32767. The default is 1521.

### Examples

snmpsubagentport 1525

---

## SSLTCPADMINPORT

The SSLTCPADMINPORT option specifies the port address on which the server TCP/IP communication driver waits for requests for Secure Sockets Layer-enabled sessions for the command line administrative client.

The following types of sessions do not use secure sockets layer (SSL):

- Server-to-server
- NDMP (Network Data Management Protocol)
- ACSLS (Automated Cartridge System Library Software)
- SNMP (Simple Network Management Protocol) subagent
- Storage agent
- Library client
- Managed server
- Event server

If the ADMINONCLIENTPORT option is set to NO, SSL-enabled sessions for the administrative client require SSLTCPADMINPORT with a port number other than that specified by the SSLTCPPORT option. The SSLTCPADMINPORT option does not affect the TCPPORT or TCPADMINPORT options and their interaction with the ADMINONCLIENTPORT option.

The TCP/IP communications driver must be enabled with COMMMETHOD TCPIP or COMMMETHOD V6TCPIP.

### Syntax

▶▶—SSLTCPADMINPort—*port\_number*————▶▶

### Parameters

*port\_number*

Specifies the port number of the server. Valid values are 1024 - 32767. There is no default.

### Examples

```
ssltcpadminport 1543
```

## SSLTCPPOINT

The SSLTCPPOINT option specifies the Secure Sockets Layer (SSL)-only port number on which the server TCP/IP communication driver waits for requests for SSL-enabled sessions for the command line backup-archive client and the command line administrative client.

The following types of sessions do not use SSL:

- Server-to-server
- NDMP (Network Data Management Protocol)
- ACSLS (Automated Cartridge System Library Software)
- SNMP (Simple Network Management Protocol) subagent
- SNMP storage agent
- SNMP library client
- SNMP managed server
- SNMP event server

If the ADMINONCLIENTPORT option is set to NO, SSL-enabled sessions for the administrative client require SSLTCPADMINPORT with a port number other than that specified by the SSLTCPPOINT option. The SSLTCPPOINT option does not affect the TCPPOINT or TCPADMINPORT options and their interaction with the ADMINONCLIENTPORT option.

The TCP/IP communications driver must be enabled with COMMMETHOD TCP/IP or COMMMETHOD V6TCP/IP.

### Syntax

►►—SSLTCPPOINT—*port\_number*—————►◄

### Parameters

*port\_number*

Specifies the port number of the server. Valid values are 1024 - 32767. There is no default.

### Examples

```
ssltcpport 1542
```

---

## TCPADMINPORT

The TCPADMINPORT option specifies the port number on which the server TCP/IP communication driver is to wait for requests for sessions other than client sessions. This includes administrative sessions, server-to-server sessions, Simple Network Management Protocol (SNMP) subagent sessions, storage agent sessions, library client sessions, managed server sessions, and event server sessions.

Using different port numbers for the options TCPPORT and TCPADMINPORT enables you to create one set of firewall rules for client sessions and another set for the other session types listed above. By using the **SESSIONINITIATION** parameter of REGISTER and UPDATE NODE, you can close the port specified by TCPPORT at the firewall, and specify nodes whose scheduled sessions will be started from the server. If the two port numbers are different, separate threads will be used to service client sessions and the session types. If you allow the two options to use the same port number (by default or by explicitly setting them to the same port number), a single server thread will be used to service all session requests.

Client sessions attempting to use the port specified by TCPADMINPORT will be terminated (if TCPPORT and TCPADMINPORT specify different ports). Administrative sessions are allowed on either port, (unless the ADMINONCLIENTPORT option is set to NO) but by default will use the port specified by TCPADMINPORT.

### Syntax

►►—TCPADMINPort—*port\_number*—◄◄

### Parameters

*port\_number*

Specifies the port number of the server. Valid values are 1024 - 32767. The default is the value of TCPPORT.

### Examples

tcpadminport 1502

---

## TCPBUFSIZE

The TCPBUFSIZE option specifies the size of the buffer used for TCP/IP send requests. During a restore, client data moves from the Tivoli Storage Manager session component to a TCP communication driver. The TCPBUFSIZE option determines if the server sends the data directly from the session buffer or copies the data to the TCP buffer. A 32 KB buffer size forces the server to copy data to its communication buffer and flush the buffer when it fills.

**Note:** This option is not related to the TCPWINDOWSIZE option.

### Syntax

►►—TCPBufsize—*kilobytes*—►►

### Parameters

*kilobytes*

Specifies the size, in kilobytes, of the buffer used for TCP/IP send requests. The range is from 1 to 32. The default is 32.

### Examples

tcpbufsize 5

---

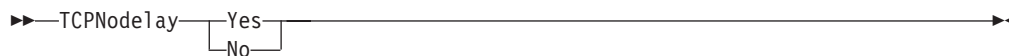
## TCPNODELAY

The TCPNODELAY option specifies whether the server disables the delay of sending successive small packets on the network.

Change the value from the default of YES only under one of these conditions:

- You are directed to change the option by your service representative.
- You fully understand the effects of the TCP Nagle algorithm on network transmissions. Setting the option to NO enables the Nagle algorithm, which delays sending small successive packets.

### Syntax



### Parameters

#### Yes

Specifies that the server allows successive small packets to be sent immediately over the network. Setting this option to YES might improve performance in some high-speed networks. The default is YES.

#### No

Specifies that the server does not allow successive small packets to be sent immediately over the network.

### Examples

```
tcpnodeLAY no
```



---

## TCPSPORT

The TCPSPORT option specifies the port number on which the server TCP/IP communication driver is to wait for requests for client sessions.

Using different port numbers for the options TCPSPORT and TCPADMINPORT enables you to create one set of firewall rules for client sessions and another set for other session types (administrative sessions, server-to-server sessions, Simple Network Management Protocol (SNMP) subagent sessions, storage agent sessions, library client sessions, managed server sessions, and event server sessions). If the two port numbers are different, separate threads will be used to service client sessions and the other session types. If you allow the two options to use the same port number (by default or by explicitly setting them to the same port number), a single server thread will be used to service all session requests.

You can change this option with the SETOPT command. When you change a port, the Tivoli Storage Manager server starts listening on the new port immediately. All current connections remain in use until closed.

### Syntax

►►—TCPSPort—*port\_number*—◄◄

### Parameters

*port\_number*

Specifies the port number of the server. Valid values are 1024 - 32767. The default value is 1500.

tcpport 1500

---

## TCPWINDOWSIZE

The TCPWINDOWSIZE option specifies, in kilobytes, the amount of receive data that can be buffered at one time on a TCP/IP connection. The sending host cannot send more data until it receives an acknowledgment and a TCP receive window update. Each TCP packet contains the advertised TCP receive window on the connection. A larger window lets the sender continue sending data, and may improve communication performance, especially on fast networks with high latency.

### Note:

- To improve backup performance, increase the TCPWINDOWSIZE on the server. To improve restore performance, increase the TCPWINDOWSIZE on the client.
- The TCP window acts as a buffer on the network.
- A window size larger than the buffer space on the network adapter might degrade throughput due to resending packets that were lost on the adapter.
- The TCPWINDOWSIZE option is not related to the TCPBUFFSIZE option nor to the send and receive buffers allocated in client or server memory.

### Syntax

►►—TCPWindowsize—*kilobytes*—◄◄

### Parameters

*kilobytes*

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. The default is 63. If you specify 0, the server uses the default window size set by the operating system. Values from 1 to 2048 indicate that the window size is in the range of 1 KB to 2 MB.

### Examples

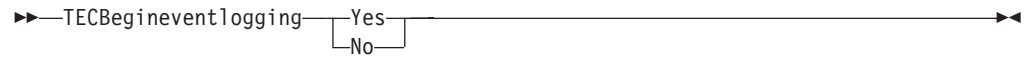
```
tcpwindowsize 8
```

---

## TECBEGINEVENTLOGGING

The TECBEGINEVENTLOGGING option specifies whether event logging for the Tivoli receiver should begin when the server starts up. If the TECHOST option is specified, TECBEGINEVENTLOGGING defaults to YES.

### Syntax



### Parameters

#### Yes

Specifies that event logging begins when the server starts up and if a TECHOST option is specified.

#### No

Specifies that event logging should not begin when the server starts up. To later begin event logging to the Tivoli receiver (if the TECHOST option has been specified), you must issue the BEGIN EVENTLOGGING command.

### Examples

```
tecbegineventlogging yes
```

---

## TECHOST

The TECHOST option specifies the host name or IP address for the Tivoli event server.

### Syntax

►►—TECHost—*host\_name*—————►◄

### Parameters

*host\_name*

Specifies the host name or IP address for the Tivoli event server.

### Examples

techost 9.114.22.345

---

## TECPORT

The TECPORT option specifies the TCP/IP port address on which the Tivoli event server is listening. This option is only required if the Tivoli event server is on a system that does not have a Port Mapper service running.

### Syntax

►►—TECPort—*port\_number*—————►◄

### Parameters

*port\_number*

Specifies the Tivoli event server port address. The value must be between 0 and 32767. This option is not required.

### Examples

tecport 1555

---

## TECUTF8EVENT

The TECUTF8EVENT option allows the Tivoli Storage Manager administrator to send information to the Tivoli Enterprise Console (TEC) server in UTF-8 data format. The default is No. You can display whether or not this option is enabled by issuing the QUERY OPTION command.

### Syntax

►►—TECUTF8event—

|     |
|-----|
| Yes |
| No  |

►►

### Parameters

#### Yes

Specifies that the Tivoli Storage Manager server will encode the TEC event into UTF-8 before issuing the event to the TEC server.

#### No

Specifies that Tivoli Storage Manager server will not encode the TEC event into UTF-8 and it will be issued to the TEC server in ASCII format.

### Examples

```
tecutf8event yes
```

## THROUGHPUTDATATHRESHOLD

The THROUGHPUTDATATHRESHOLD option specifies a throughput threshold that a client session must reach to prevent being cancelled after the time threshold is reached.

This option is used in conjunction with the THROUGHPUTTIMETHRESHOLD server option, which sets the value for the time threshold plus the media wait time. The time threshold starts when the client begins sending data to the server for storage (as opposed to setup or session housekeeping data).

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—THROUGHPUTDatathreshold— *kilobytes\_per\_second* —►►

### Parameters

*kilobytes\_per\_second*

Specifies the throughput that client sessions must achieve to prevent cancellation after THROUGHPUTTIMETHRESHOLD minutes have elapsed.

This threshold does not include time spent waiting for media mounts. A value of 0 prevents examining client sessions for insufficient throughput. Throughput is computed by adding send and receive byte counts and dividing by the length of the session. The length does not include time spent waiting for media mounts and starts at the time a client sends data to the server for storage. The default is 0. The minimum value is 0; the maximum is 99999999.

### Examples

Specify that the server is to wait until 90 minutes plus the media wait time after a session has started sending data before storage examines it as a candidate for cancellation due to low throughput. If a session is not achieving 50 KB per second in transfer rates, it will be cancelled.

```
throughputtimethreshold 90
Throughputdatathreshold 50
```

---

## THROUGHPUTTIMETHRESHOLD

The THROUGHPUTTIMETHRESHOLD option specifies the time threshold for a session after which it may be cancelled for low throughput.

You can update this server option without stopping and restarting the server by using the SETOPT command. See “SETOPT (Set a server option for dynamic update)” on page 956.

### Syntax

►►—THROUGHPUTtimethreshold—*minutes*—◄◄

### Parameters

*minutes*

Specifies the threshold for examining client sessions and cancelling them if the data throughput threshold is not met (see the THROUGHPUTDATATHRESHOLD server option). This threshold does not include time spent waiting for media mounts. The time threshold starts when a client begins sending data to the server for storage (as opposed to setup or session housekeeping data). A value of 0 prevents examining client sessions for low throughput. The default is 0. The minimum value is 0; the maximum is 99999999.

### Examples

Specify that the server is to wait until 90 minutes plus the media wait time after a session has started sending data before examining it as a candidate for cancellation. If a session is not achieving 50 thousand bytes per second in transfer rates, it will be cancelled.

```
throughputtimethreshold 90  
Throughputdatathreshold 50
```



## TXNGROUPMAX

The TXNGROUPMAX option specifies the number of objects that are transferred as a group between a client and the server between transaction commit points. The minimum value is 4 objects and the maximum value is 65000 objects. The default value is 4096 objects. The objects transferred are actual files, directories, or both. The server counts each file or directory as one object.

It is possible to affect the performance of client backup, archive, restore, and retrieve operations by using a larger value for this option:

1. If you increase the value of the TXNGROUPMAX option by a large amount, watch for possible effects on the recovery log. A larger value for the TXNGROUPMAX option can result in increased utilization of the recovery log, as well as an increased length of time for a transaction to commit. If the effects are severe enough, they can lead to problems with operation of the server.
2. Increasing the value of the TXNGROUPMAX option can improve throughput for operations storing data directly to tape, especially when storing a large number of objects. However, a larger value of the TXNGROUPMAX option can also increase the number of objects that must be resent in the case where the transaction is stopped because an input file changed during backup, or because a new storage volume was required. The larger the value of the TXNGROUPMAX option, the more data must be resent.
3. Increasing the TXNGROUPMAX value will affect the responsiveness of stopping the operation and the client may have to wait longer for the transaction to complete.

You can override the value of this option for individual client nodes. See the TXNGROUPMAX parameter in “REGISTER NODE (Register a node)” on page 837 and “UPDATE NODE (Update node attributes)” on page 1071.

This option is related to the TXNBYTELIMIT option in the client options file. TXNBYTELIMIT controls the number of bytes, as opposed to the number of objects, that are transferred between transaction commit points. At the completion of transferring an object, 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 objects transferred.

### Syntax

►►—TXNGroupmax—*number\_of\_objects*—————►►

### Parameters

*number\_of\_objects*

Specifies a number from 4 to 65000 for the maximum number of objects per transaction. The default is 4096.

### Examples

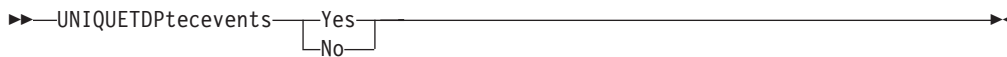
txngroupmax 512

---

## UNIQUETDPTECEVENTS

The UNIQUETDPTECEVENTS option generates a unique Tivoli Enterprise Console (TEC) event class for each individual Tivoli Storage Manager message, including client, server, and Tivoli Data Protection (TDP) agent messages. The default is No.

### Syntax



### Parameters

#### Yes

Specifies that unique TDP messages are sent to the TEC event server.  
Dynamically sets UNIQUETEEvents to YES.

#### No

Specifies that general messages are sent to the TEC event server.

### Examples

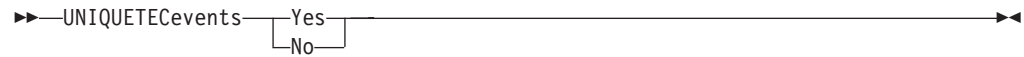
```
uniquetdpcevents yes
```

---

## UNIQUETECEVENTS

The UNIQUETECEVENTS option generates a unique Tivoli Enterprise Console (TEC) event class for each individual Tivoli Storage Manager message. The default is No.

### Syntax



### Parameters

#### Yes

Specifies that unique messages are sent to the TEC event server.

#### No

Specifies that general messages are sent to the TEC event server.

### Examples

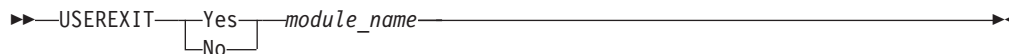
```
uniquecevents yes
```

---

## USEREXIT

The USEREXIT option specifies a user-defined exit that will be given control to manage an event.

### Syntax



### Parameters

#### Yes

Specifies that event logging to the user exit receiver begins automatically at server startup.

#### No

Specifies that event logging to the user exit receiver does not begin automatically at server startup. When this parameter has been specified, you must begin event logging manually by issuing the BEGIN EVENTLOGGING command.

#### *module\_name*

Specifies the module name of the user exit.

This is the name of a shared library containing the exit. The module name can be either a fully qualified path name or just the module name itself. If it is just the module name, it is loaded from the current directory.

### Examples

```
userexit yes fevent.exit
```

---

## VERBCHECK

The VERBCHECK option specifies that the server will do additional error checking on the structure of commands sent by the client. This option should only be enabled when the client sends incorrectly formed requests to the server, causing the server to crash. When this option is enabled, you will get a protocol error instead of a server crash.

### Syntax

►►—VERBCHECK—◄◄

### Parameters

None

### Examples

Enable additional error checking for commands sent by the client:

```
verbcheck
```

---

## VOLUMEHISTORY

The VOLUMEHISTORY option specifies the name of files to be automatically updated whenever server sequential volume history information is changed. There is no default for this option.

You can include one or more VOLUMEHISTORY options in the server options file. When you use multiple VOLUMEHISTORY options, the server automatically updates and stores a backup copy of the volume history information in each file you specify.

### Syntax

►►—VOLUMEHistory—*file\_name*—————►◄

### Parameters

*file\_name*

Specifies the name of the file where you want the server to store a backup copy of the volume history information that it collects.

### Examples

volumehistory volhist.out

---

## Chapter 4. Server utilities

Use server utilities to perform special tasks on the server while the server is not running.

*Table 401. Server utilities*

| Utility   | Description  |
|---|--|
| "DSMSERV (Start the server)" on page 1258   | Starts the server.   |
| "The server startup script: rc.dsmserv" on page 1259                                      | Automatically starts a server instance.                            |
| "DSMSERV DISPLAY DBSPACE (Display information about database storage space)" on page 1260 | Displays information about storage space defined for the database. |
| "DSMSERV DISPLAY LOG (Display recovery log information)" on page 1261                     | Displays information about recovery log storage space.             |
| "DSMSERV FORMAT (Format the database and log)" on page 1263                               | Initializes the database and recovery log.                         |
| "DSMSERV INSERTDB (Move a server database into an empty database)" on page 1265           | Inserts a server database into a new Version 6 database.           |
| "DSMSERV LOADFORMAT (Format a database)" on page 1267                                     | Formats an empty database.   |
| "DSMSERV REMOVEDB (Remove a database)"<br>  on page 1269                                  | Removes an IBM Tivoli Storage Manager database.                    |
| "DSMSERV RESTORE DB (Restore the database)" on page 1270                                  | Restores an IBM Tivoli Storage Manager database.                   |
| "DSMULOG (Capture a message to a user log file)" on page 1276                             | Increases the maximum transfer length for writing to tape.         |

## DSMSERV (Start the server)

Use this utility to start the Tivoli Storage Manager server. You might be asked to specify the RUNFILE parameter after you install maintenance or a new version of Tivoli Storage Manager.

### Syntax

```

DSMSERV [-u user_name] [-i instance_dir] [-o options_file]
        [-noexpire] [-quiet] RUNFILE file_name

```

### Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server.

**-o** *options\_file*

Specifies an options file to use.

**-noexpire**

Specifies that expiration processing is suppressed when starting the server.

**-quiet**

Specifies that messages to the console are suppressed.

**RUNFILE** *file\_name*

Specifies the name of a text file to be run on the server. The file contains a list of Tivoli Storage Manager server commands.

**Attention:** Whenever the RUNFILE parameter is used, the server halts when processing is complete. You must restart the server by using the DSMSERV utility.

### Example: Start the server

Start the server for normal operation.

```
/opt/tivoli/tsm/server/bin/dsmserv
```

### Example: Load the sample script

Load the sample script file that is provided with the server.

```
dsmserv runfile scripts.smp
```



---

## The server startup script: rc.dsmserv

The startup script can be used in your system startup to automatically start a server instance under a specific user ID.

### Syntax

```
►► rc.dsmserv [-u user_name] [-U user_name] [-i instance_dir] ◀◀
```

### Parameters

**-u** *user\_name*

Specifies the Tivoli Storage Manager instance user ID for which the environment is set up. The server will run under this user ID.

**-U** *user\_name*

Specifies the Tivoli Storage Manager instance user ID for which the environment is set up. The server will run under the user ID of the invoker of the command.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server.

## DSMSERV DISPLAY DBSPACE (Display information about database storage space)

Use this utility to display information about storage space that is defined for the database. The output of this utility is the same as the output of QUERY DBSPACE, but you can use this utility when the server is not running.

### Syntax

```

DSMSERV [-u user_name] [-i instance_dir] [-o options_file]
        [-noexpire] [-quiet] DISPlay DBSPace

```

### Parameters

- u *user\_name***  
Specifies a user name to switch to before initializing the server.
- i *instance\_dir***  
Specifies an instance directory to use. This becomes the current working directory of the server.
- o *options\_file***  
Specifies an options file to use.
- noexpire**  
Specifies that expiration processing is suppressed when starting.
- quiet**  
Specifies that messages to the console are suppressed.

### Example: Display database space information

Display information about database storage space. See “Field descriptions” for details about the information shown in the output. Issue the command.

```
dsmserv display dbspace
```

| Location  | Total Space (MB) | Used Space (MB) | Free Space (MB) |
|-----------|------------------|-----------------|-----------------|
| /tsmdb001 | 46,080.00        | 20,993.12       | 25,086.88       |
| /tsmdb002 | 46,080.00        | 20,992.15       | 25,087.85       |

### Field descriptions

#### Location

The directory or path that is used for storing the database

#### Total Space (MB)

The total number of megabytes in the location

#### Used Space (MB)

The number of megabytes in use in the location

#### Free Space (MB)

The space remaining in the file system where the path is located

## DSMSERV DISPLAY LOG (Display recovery log information)

Use this utility to display information about recovery logs including the active log, the mirror for the active log, the failover directory for the archive log, and the overflow location for logs. Use this utility when the server is not running.

### Syntax

```

DSMSERV -u user_name -i instance_dir -o options_file
DISPLAY LOG -noexpire -quiet

```

### Parameters

- u *user\_name***  
Specifies a user name to switch to before initializing the server.
- i *instance\_dir***  
Specifies an instance directory to use. This becomes the current working directory of the server.
- o *options\_file***  
Specifies an options file to use.
- noexpire**  
Specifies that expiration processing is suppressed when starting.
- quiet**  
Specifies that messages to the console are suppressed.

### Examples: Display recovery log information

Display information about the recovery logs. See “Field descriptions” for details about the information shown in the output.

```
dsmserv display log
```

```

Total Space(MB): 38,912
Used Space(MB): 401.34
Free Space(MB): 38,358.65
Active Log Directory: /activelog
Mirror Log Directory: /mirrorlog
Archive Failover Log Directory: /archfailoverlog
Archive Log Directory: /archivelog

```

### Field descriptions

#### Total Space

Specifies the maximum size of the active log.

#### Used Space

Specifies the total amount of active log space currently used in the database, in megabytes.

#### Free Space

Specifies the amount of active log space in the database that is not being used by uncommitted transactions, in megabytes.

## DSMSERV DISPLAY LOG

### **Active Log Directory**

Specifies the location where active log files are stored. When you change the active log directory, the server moves all archived logs to the archive log directory and all active logs to a new active log directory.

### **Mirror Log Directory**

Specifies the location where the mirror for the active log is maintained.

### **Archive Failover Log Directory**

Specifies the location in which the server saves archive logs if the logs cannot be archived to the archive log destination.

## DSMSERV FORMAT (Format the database and log)

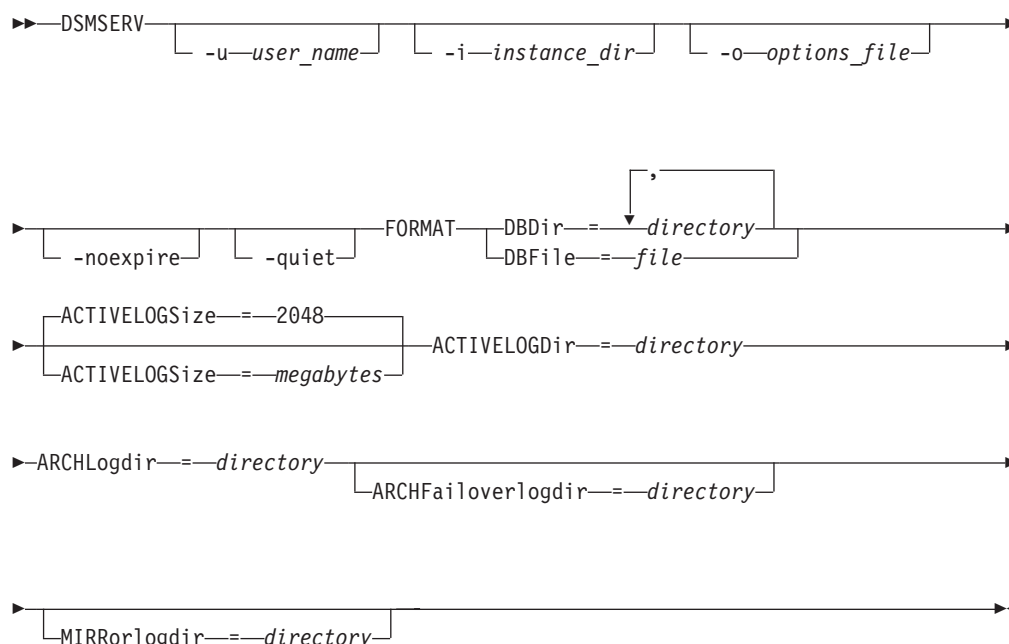
Use the DSMSERV FORMAT utility to initialize the server database and recovery log. No other server activity is allowed while initializing the database and recovery log.

The directories that are specified in this utility should be on fast, reliable storage. Do not place the directories on file systems that might run out of space. If certain directories (for example, the active log directory) become unavailable or full, the server stops.

Before you issue the DSMSERV FORMAT command, log on to the system as the server instance owner. If you are running under something other than the English regional locale, set the DB2CODEPAGE system environment variable to 819:

```
# db2set -i tsminst1 DB2CODEPAGE=819
```

### Syntax



### Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server. This parameter is optional.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server. This parameter is optional.

**-o** *options\_file*

Specifies an options file to use. This parameter is optional.

**-noexpire**

Specifies that expiration processing is suppressed when starting. This parameter is optional.

## DSMSERV FORMAT

### **-quiet**

Specifies that messages to the console are suppressed. This parameter is optional.

### **DBDir**

Specifies the relative path names of one or more directories that are used to store database objects. Directory names must be separated by commas but without spaces. You can specify up to 128 directory names. You must specify either the DBDIR or the DBFILE parameter.

### **DBFile**

Specifies the name of a file that contains the relative path names of one or more directories that are used to store database objects. Each directory name must be on a separate line in the file. You can specify up to 128 directory names. You must specify either the DBDIR or the DBFILE parameter.

### **ACTIVELOGSize**

Specifies the size of the active log in megabytes. This parameter is optional. The minimum value is 2048 MB (2 GB); the maximum is 131072 MB (128 GB). If you specify an odd number, the value is rounded up to the next even number. The default is 2048 MB.

### **ACTIVELOGDir (Required)**

Specifies the directory in which the Tivoli Storage Manager server writes and stores active log files. There is only one active log location. The name must be a fully qualified directory name. The directory must already exist, it must be empty, and it must be accessible by the user ID of the database manager. The maximum number of characters is 175.

### **ARCHLogdir (Required)**

Specifies the directory for the archive log files. The name must be a fully qualified directory name. The maximum number of characters is 175.

### **ARCHFailoverlogdir**

Specifies the directory to be used as an alternate storage location if the ARCHLOGDIR directory is full. This parameter is optional. The maximum number of characters is 175.

### **MIRRorlogdir**

Specifies the directory in which the server mirrors the active log (those files in the ACTIVELOGDIR directory). This parameter is optional. The directory must be a fully qualified directory name. The maximum number of characters is 175.

### **Example: Format a database**

```
dsmserv format dbdir=/tsmdb001 activelogsize=8192  
activelogdir=/active log archlogdir=/archivelog  
archfailoverlogdir=/archfaillog mirrorlogdir=/mirrorlog
```

## DSMSERV INSERTDB (Move a server database into an empty database)

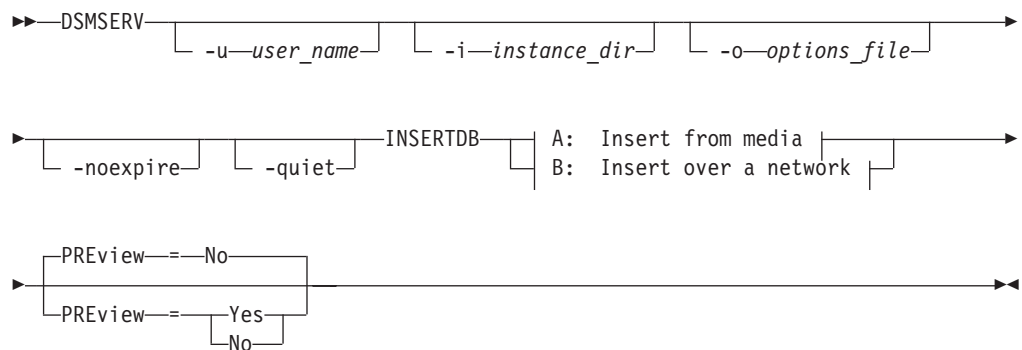
Use the DSMSERV INSERTDB utility to move a server database into a new V6.1 database. The database can be extracted from the original server and inserted into a new database on the new server by using a network connection between the original server and the new server. The database can also be inserted from media that contains the extracted database.

### Requirements for insertion using media

Before you run the utility to insert the server database into an empty database, ensure that your system meets the following requirements.

- The manifest file from the DSMUPGRD EXTRACTDB operation must be available.
- The server options file must contain an entry for the device configuration file.
- The device configuration file must have information about the device class that is specified in the manifest file.
- The media that contains the extracted database must be available to the V6.1 server. The device must be physically attached to the system, and the permissions must be set to grant access to the media for the user ID that owns the V6.1 server instance.

### Syntax



#### A: Insert from media:



#### B: Insert over a network:



## Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server. This parameter is optional.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server. This parameter is optional.

**-o** *options\_file*

Specifies an options file to use. This parameter is optional.

**-noexpire**

Specifies that expiration processing is suppressed when starting. This parameter is optional.

**-quiet**

Specifies that messages to the console are suppressed. This parameter is optional.

## DEVclass

Specifies a sequential-access device class. You can specify any device class except for the DISK device class. The definition for the device class must exist in the device configuration file.

This parameter is optional and is used only when the database that you want to insert into the empty V6.1 database was extracted to media. If the database is on media and you do not specify a device class, the device class that is identified in the manifest file is used.

**Restriction:** You cannot use a device class with a device type of NAS or CENTERA.

## MANifest

Specifies the location of the manifest file. Use a fully qualified file name, or place in a local directory. For example: `./manifest.txt`

This parameter is required when the database that you want to insert into the empty V6.1 database was extracted to media.

## SESSWait

Specifies the number of minutes that the V6.1 server waits to be contacted by the original server. The default value is 60 minutes.

Use this parameter only if the data that you want to insert into the empty V6.1 database is to be transmitted from the source server using a network connection.

## PREview

Specifies whether to preview the insertion operation. This parameter is optional. The default value is **NO**.

Use the **PREVIEW=YES** parameter to test a database. When you use this parameter, the operation includes all steps of the process, except for the actual insertion of data into the new database. When you preview the insertion operation, you can quickly verify that the source database is readable. You can also identify any data constraint violations that might prevent an upgraded database from being put into production.



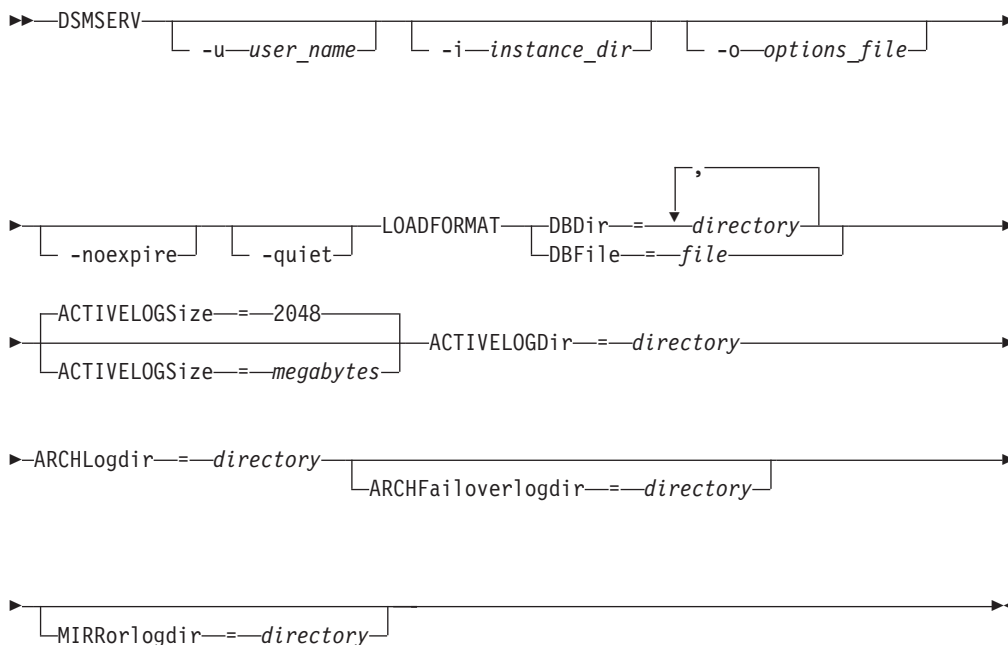
## DSMSERV LOADFORMAT (Format a database)

Use the DSMSERV LOADFORMAT utility when upgrading from Version 5. The utility formats an empty database in preparation for inserting an extracted database into the empty database.

Before you issue the DSMSERV LOADFORMAT command, log on to the system as the server instance owner. If you are running under something other than the English regional locale, set the DB2CODEPAGE system environment variable to 819:

```
# db2set -i tsminst1 DB2CODEPAGE=819
```

### Syntax



### Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server. This parameter is optional.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server. This parameter is optional.

**-o** *options\_file*

Specifies an options file to use. This parameter is optional.

**-noexpire**

Specifies that expiration processing is suppressed when starting. This parameter is optional.

**-quiet**

Specifies that messages to the console are suppressed. This parameter is optional.

### **DBDir**

Specifies the relative path names of one or more directories that are used to store database objects. Directory names must be separated by commas but without spaces. You can specify up to 128 directory names. You must specify either the DBDIR or the DBFILE parameter.

### **DBFile**

Specifies the name of a file that contains the relative path names of one or more directories that are used to store database objects. Each directory name must be on a separate line in the file. You can specify up to 128 directory names. You must specify either the DBDIR or the DBFILE parameter.

### **ACTIVELOGSize**

Specifies the size of the active log in megabytes. This parameter is optional. The minimum value is 2048 MB (2 GB); the maximum is 131072 MB (128 GB). If you specify an odd number, the value is rounded up to the next even number. The default is 2048 MB.

### **ACTIVELOGDir (Required)**

Specifies the directory in which the Tivoli Storage Manager server writes and stores active log files. There is only one active log location. The name must be a fully qualified directory name. The directory must already exist, it must be empty, and it must be accessible by the user ID of the database manager. The maximum number of characters is 175.

### **ARCHLogdir (Required)**

Specifies the directory for the archive log files. The name must be a fully qualified directory name. The maximum number of characters is 175.

### **ARCHFailoverlogdir**

Specifies the directory to be used as an alternate storage location if the ARCHLOGDIR directory is full. This parameter is optional. The maximum number of characters is 175.

### **MIRRORlogdir**

Specifies the directory in which the server mirrors the active log (those files in the ACTIVELOGDIR directory). This parameter is optional. The directory must be a fully qualified directory name. The maximum number of characters is 175.

### **Example: Format a database**

```
dsmserv format dbdir=/tsmdb001 activelogsize=8192  
activelogdir=/active log archlogdir=/archive log  
archfailoverlogdir=/archfail log mirrorlogdir=/mirror log
```

## DSMSERV REMOVEDB (Remove a database)

Use the DSMSERV REMOVEDB utility to remove a Tivoli Storage Manager server database.

Use this command to remove a Tivoli Storage Manager server database. This command deletes all user data and log files, as well as any backup and restore history for the database. If the log files are needed for a roll-forward recovery after a restore operation, these files should be saved before you issue this command.

**Note:** The Tivoli Storage Manager server must be halted before you issue this command.

### Syntax

```

DSMSERV [-u user_name] [-i instance_dir] [-o options_file]
        [-noexpire] [-quiet] REMOVEDB database_name

```

### Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server.

**-o** *options\_file*

Specifies an options file to use.

**-noexpire**

Specifies that expiration processing is suppressed when starting.

**-quiet**

Specifies that messages to the console are suppressed.

*database\_name*

The database name that was entered during installation. If the database was formatted manually, then this is the database name parameter in the DSMSERV FORMAT or DSMSERV LOADFORMAT utility. This database name can also be found in `dsmserv.opt` file. This parameter is required.

### Example: Remove a database

Remove the Tivoli Storage Manager server database TSMDB1 and all of its references.

```
dsmserv removedb TSMDB1
```

---

### DSMSERV RESTORE DB (Restore the database)

Use this utility to restore a database by using a database backup.

Use this utility for the following tasks:

- “DSMSERV RESTORE DB (Restore a database to its most current state)” on page 1271
- “DSMSERV RESTORE DB (Restore a database to a point-in-time)” on page 1273

Before you perform any of these operations, you should be familiar with the procedures as they are described in the *Administrator's Guide*.

The restore operation uses database backups created with the BACKUP DB command.

**Important:** After a point-in-time restore operation, issue the AUDIT VOLUME command to audit all DISK volumes and resolve any inconsistencies between database information and storage pool volumes. Before restoring the database, examine the volume history file to find out about any sequential access storage pool volumes that were deleted or reused since the point in time to which the database was restored.

## DSMSERV RESTORE DB (Restore a database to its most current state)

Use the DSMSERV RESTORE DB utility to restore a database to its most current state under certain conditions.

The following conditions must be met:

- An intact volume history file is available.
- The recovery logs are available.
- A device configuration file with the applicable device information is available.

Tivoli Storage Manager requests volume mounts to load the most recent backup series and then uses the recovery logs to update the database to its most current state.

Snapshot database backups cannot be used to restore a database to its most current state.

### Syntax

```

DSMSERV [-u user_name] [-i instance_dir] [-o options_file]
        [-noexpire] [-quiet] RESTORE DB [RECOVdir=directory]
        [ACTIVELOGDir=directory] [ON=target_directory_file]

```

### Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server.

**-o** *options\_file*

Specifies an options file to use.

**-noexpire**

Specifies that expiration processing is suppressed when starting.

**-quiet**

Specifies that messages to the console are suppressed.

**RECOVdir**

Specifies a directory in which to store recovery log information from the database backup media. This directory must have enough space to hold this transaction recovery information and must be an empty directory. If this parameter is not specified, the default is to the directory specified by one of the following parameters in the DSMSERV FORMAT or DSMSERV LOADFORMAT utility:

- ARCHFAILOVERLOGDIR, if specified
- ARCHLOGDIR, if ARCHFAILOVERLOGDIR is not specified

## DSMSERV RESTORE DB

### ACTIVELOGDir

Specifies a directory in which to store the log files used to track the active database operations. This directory needs to be specified only if the intent is to switch to an active log directory different from the one that had already been configured.

### On

Specifies a file listing the directories to which the database will be restored. Specify each directory on a separate line in the file. For example, the ON parameter specifies the restorelist.txt file, which contains the following list:

```
/tsmdb001  
/tsmdb002  
/tsmdb003
```

If this parameter is not specified, the original directories that were recorded in the database backup are used.

### Example: Restore the database to its most current state

Restore the database to its most current state using the already configured active log directory.

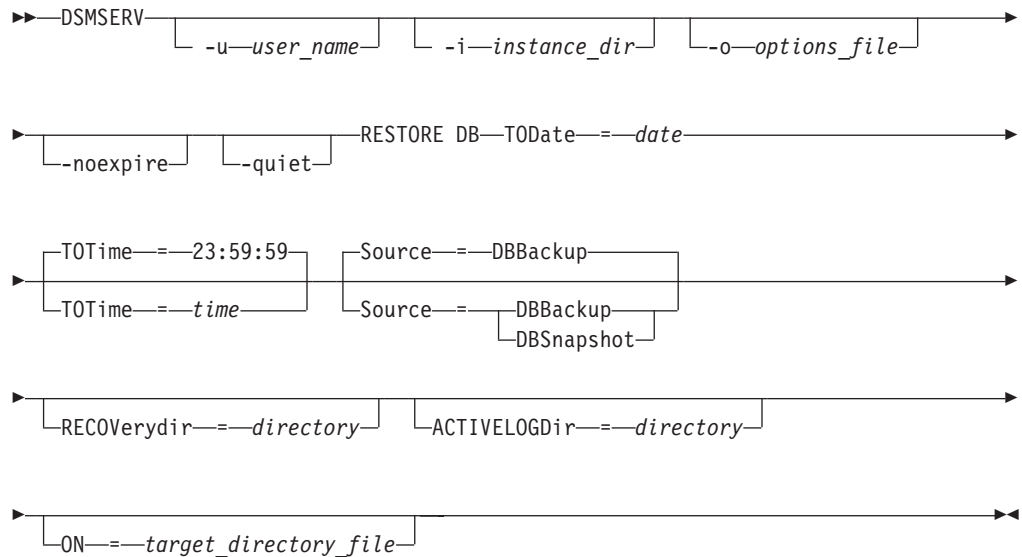
```
dsmserv restore db
```

## DSMSERV RESTORE DB (Restore a database to a point-in-time)

To restore a database to a point in time a volume history file and a device configuration file must be available.

Full and incremental database backups or snapshot database backups can be used to restore a database to a point in time.

### Syntax



### Parameters

**-u** *user\_name*

Specifies a user name to switch to before initializing the server.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server.

**-o** *options\_file*

Specifies an options file to use.

**-noexpire**

Specifies that expiration processing is suppressed when starting.

**-quiet**

Specifies that messages to the console are suppressed.

**TODate (Required)**

Specifies the date to which to restore the database. Possible values are:

*MM/DD/YYYY*

Specifies that you want to restore a database using the last backup series that was created before this specified date.

**TODAY**

Specifies that you want to restore a database using the most recent backup series that was created before today.

### **TODAY**-*numdays* **or** -*numdays*

Specifies that you want to restore a database using the most recent backup series that was created the specified number of days before the current date.

### **TOTime**

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 the specified time on the date that is specified on the TODATE parameter.

### **NOW**

Specifies that you want to restore the database using a backup series created on or before the current time on the date that is specified on the TODATE parameter.

For example, if you issue the DSMSERV RESTORE DB utility at 9:00 with TOTIME=NOW, the database is restored using the last backup series created on or prior to 9:00 on the date that is specified on the TODATE parameter.

### **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 date that is specified on the TODATE parameter.

For example, if you issue the DSMSERV RESTORE DB utility at 9:00 with TOTIME=NOW-3:30 or TOTIME+=3:30, the database is restored using the last backup series created on or prior to 5:30 on the date that is specified on the TODATE parameter.

### **Source**

Specifies whether the database is restored using either database full and incremental backup volumes or snapshot database volumes. This parameter is optional. The default value is DBBackup. Possible values are:

#### **DBBackup**

Specifies that the database is restored as follows:

1. Reads the volume history file to locate the database full and incremental backup volumes that are needed.
2. Requests mounts and loads the data from the database full and incremental backup volumes as required to restore the database volume to the specified time.

#### **DBSnapshot**

Specifies that the database is restored as follows:

1. Reads the volume history file to locate the snapshot database volumes that are needed,
2. Requests mounts and loads data from snapshot database volumes as required to restore the volume to the specified time.

#### **RECOVdir**

Specifies a directory in which to store recovery log information from the database backup media. This log information is used to establish transaction consistency of the server database as part of the recovery processing. This directory must have enough space to hold this transaction



recovery information and must be an empty directory. If this parameter is not specified, the default is the directory that is specified by one of the following parameters in the DSMSERV FORMAT or DSMSERV LOADFORMAT utility:

- ARCHFAILOVERLOGDIR, if specified
- ARCHLOGDIR, if ARCHFAILOVERLOGDIR is not specified

**ACTIVELOGDir**

Specifies a directory in which to store the log files that are used to track the active database operations. Specify this directory only if the intent is to switch to an active log directory that is different from the one that had already been configured.

**On**

Specifies a file listing the directories to which the database will be restored. Specify each directory on a separate line in the file. For example, the ON parameter specifies the restorelist.txt file, which contains the following list:

```
/tsmdb001  
/tsmdb002  
/tsmdb003
```

If this parameter is not specified, the original directories that were recorded in the database backup are used.

**Example: Restore the database to a specific time**

Restore the database to its state on May 12, 2009 at 2:25 PM.

```
dsmserv restore db todate=05/12/2009 totime=14:45
```

## DSMULOG (Capture a message to a user log file)

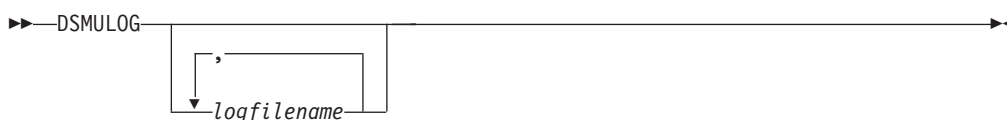
Use this command to capture Tivoli Storage Manager server console messages to a user log file. You can specify that IBM Tivoli Storage Manager write messages to more than one user log file.

You can start this command with the ADSMSTART shell script, which is provided as part of the IBM Tivoli Storage Manager AIX server package. If you do not specify fully qualified filenames for the user log files in the ADSMSTART shell script the user log files are created in the directory where ADSMSTART is started. By editing the ADSMSTART shell script, where the command is started, you can manage the amount of space and define the number of server messages written to one or more user log files.

For more information refer to the *Administrator's Guide*.

**Important:** Do not place the user logs in the /usr/lpp file system because space constraints in the file system can prevent the server from starting.

### Syntax



### Parameters

#### *logfilename* (Required)

Specifies the name of one or more user log files to which IBM Tivoli Storage Manager writes server console messages.

### Example: Setup a script to capture messages to a user log file

Set up and invoke the DSMULOG command to rotate through the user logs on a daily basis by completing the following procedure.

1. Change to the server bin directory:  

```
cd /usr/lpp/admserv/bin
```
2. Copy ADSMSTART.SMP to ADSMSTART.  

```
cp adsmstart.smp ./adsmstart
```
3. Edit ADSMSTART to specify the user log files to capture messages on a daily basis.  

```
dsmulog /u/admin/log1 /u/admin/log2 /u/admin/log3
```

---

## Appendix A. Return codes for use in IBM Tivoli Storage Manager scripts

You can write IBM Tivoli Storage Manager scripts that use return codes to determine how script processing proceeds. The return codes can be one of three severities: OK, WARNING, ERROR.

IBM Tivoli Storage Manager scripts use the symbolic return code for processing, not the numeric value. The administrative client displays the numeric values when a command is run. The return codes are shown in the following table.

*Table 402. Return codes*

| Return code     | Severity | Numeric value | Description  |
|-----------------|----------|---------------|--|
| RC_OK           | OK       | 0             | Command completed successfully   |
| RC_NOTFOUND     | WARNING  | 11            | Returned by a QUERY or SQL SELECT command when no objects are found that match specifications                |
| RC_UNKNOWN      | ERROR    | 2             | Command is not found; not a known command  |
| RC_SYNTAX       | ERROR    | 3             | Command is valid, but one or more parameters were not specified correctly                                    |
| RC_ERROR        | ERROR    | 4             | An internal server error prevented the command from successfully completing                                  |
| RC_NOMEMORY     | ERROR    | 5             | The command could not be completed because of insufficient memory on the server                              |
| RC_NOLOG        | ERROR    | 6             | The command could not be completed because of insufficient recovery log space on the server                  |
| RC_NODB         | ERROR    | 7             | The command could not be completed because of insufficient database space on the server                      |
| RC_NOSTORAGE    | ERROR    | 8             | The command could not be completed because of insufficient storage space on the server                       |
| RC_NOAUTH       | ERROR    | 9             | The administrator is not authorized to issue the command   |
| RC_EXISTS       | ERROR    | 10            | The command failed because the specified object already exists on the server                                 |
| RC_INUSE        | ERROR    | 12            | The command failed because the object to be operated upon was in use   |
| RC_ISREFERENCED | ERROR    | 13            | The command failed because the object to be operated upon is still referenced by some other server construct |
| RC_NOTAVAILABLE | ERROR    | 14            | The command failed because the object to be operated upon is not available                                   |
| RC_IOERROR      | ERROR    | 15            | The command failed because an input/output (I/O) error was encountered on the server                         |

Table 402. Return codes (continued)

| Return code   | Severity | Numeric value | Description  |
|---------------|----------|---------------|--|
| RC_NOTXN      | ERROR    | 16            | The command failed because a database transaction failed on the server               |
| RC_NOLOCK     | ERROR    | 17            | The command failed because a lock conflict was encountered in the server database    |
| RC_NOTHREAD   | ERROR    | 19            | The command could not be completed because of insufficient memory on the server      |
| RC_LICENSE    | ERROR    | 20            | The command failed because the server is not in compliance with licensing            |
| RC_INVDEST    | ERROR    | 21            | The command failed because a destination value was invalid                           |
| RC_IFILEOPEN  | ERROR    | 22            | The command failed because an input file that was needed could not be opened         |
| RC_OFILEOPEN  | ERROR    | 23            | The command failed because it could not open a required output file                  |
| RC_OFILEWRITE | ERROR    | 24            | The command failed because it could not successfully write to a required output file |
| RC_INVADMIN   | ERROR    | 25            | The command failed because the administrator was not defined                         |
| RC_SQLERROR   | ERROR    | 26            | An SQL error was encountered during a SELECT statement query                         |

---

## Appendix B. Accessibility features for Tivoli Storage Manager

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features of Tivoli Storage Manager are described in this topic.

### Accessibility features

The following list includes the major accessibility features in Tivoli Storage Manager:

- Keyboard-only operation
- Interfaces that are commonly used by screen readers
- Keys that are discernible by touch but do not activate just by touching them
- Industry-standard devices for ports and connectors
- The attachment of alternative input and output devices
- User documentation provided in HTML and PDF format. Descriptive text is provided for all documentation images.

The Tivoli Storage Manager Information Center, and its related publications, are accessibility-enabled.

### Keyboard navigation

Tivoli Storage Manager follows AIX operating system conventions for keyboard navigation and access.

### Vendor software

Tivoli Storage Manager includes certain vendor software that is not covered under the IBM license agreement. IBM makes no representation about the accessibility features of these products. Contact the vendor for the accessibility information about its products.

### Related accessibility information

You can view the publications for Tivoli Storage Manager in Adobe® Portable Document Format (PDF) using the Adobe Acrobat Reader. You can access these or any of the other documentation PDFs at the IBM Publications Center at <http://www.ibm.com/shop/publications/order/>.

### IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center at <http://www.ibm.com/able>.



---

## Notices

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

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

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

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

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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  
2Z4A/101  
11400 Burnet Road  
Austin, TX 78758  
U.S.A.*

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

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

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

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.



---

## Trademarks

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

Adobe is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

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



---

## Glossary

This glossary includes terms and definitions for IBM Tivoli Storage Manager.

To view glossaries for other IBM products, go to <http://www.ibm.com/software/globalization/terminology/>.

The following cross-references are used in this glossary:

- *See* refers the reader from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- *See also* refers the reader to a related or contrasting term.

### A

#### **absolute mode**

In storage management, a backup copy-group mode that specifies that a file is considered for incremental backup even if the file has not changed since the last backup. See also *modified mode*.

#### **access control list (ACL)**

In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights. For example, an access control list is associated with a file that identifies the users who can access that file and their access rights.

#### **access mode**

An attribute of a storage pool or a storage volume that specifies whether the server can write to or read from the storage pool or storage volume. 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.

#### **acknowledgment**

The transmission of acknowledgment characters as a positive response to a data transmission.

**ACL** See *access control list*.

#### **activate**

To validate the contents of a policy set and then make it the active policy set.

#### **active-data pool**

A named set of storage pool volumes that contain only active versions of client backup data.

#### **active file system**

A file system to which space management has been added. With space management, tasks for an active file system include automatic migration, reconciliation, selective migration, and recall. Contrast with *inactive file system*.

#### **active policy set**

The activated policy set that contains the policy rules in use by all client nodes that are assigned to the policy domain. See also *policy domain* and *policy set*.

#### **active version**

The most recent backup copy of a file stored. The active version of a file cannot be deleted until a backup process detects that the user has either replaced the file with a newer version or has deleted the file from the file server or workstation. Contrast with *inactive version*.

#### **activity log**

A log that records normal activity messages that are 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.

#### **adaptive subfile backup**

A type of backup that sends only changed portions of a file to the server, instead of sending the entire file. Adaptive subfile backup reduces network traffic and increases the speed of the backup.

#### **administrative client**

A program that runs on a file server, workstation, or mainframe that administrators use to control and monitor the Tivoli Storage Manager server. 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**

See *privilege class*.

**administrative session**

A period of time during which an administrator user ID communicates with a server to perform administrative tasks. Contrast with *client node session*.

**administrator**

A user who is registered to the server as an administrator, and who is authorized to perform tasks and issue commands through the assignment of an administrative privilege class.

**Advanced Program-to-Program Communication (APPC)**

An implementation of the SNA LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

**agent node**

A client node that has been granted proxy authority to perform operations on behalf of another client node, which is the target node.

**aggregate**

An object, stored in one or more storage pools, consisting of a group of logical files that are packaged together. See also *logical file* and *physical file*.

**aggregate data transfer rate**

A performance statistic that indicates the average number of bytes that were transferred per second while processing a given operation.

**APPC** See *Advanced Program-to-Program Communication*.

**application client**

A program that is installed on a system to protect an application. The Tivoli Storage Manager server provides backup services to an application client.

**archive**

To copy programs, data, or files to another storage media, usually for long-term storage or security. Contrast with *retrieve*.

**archive copy**

A file or group of files that was archived to server storage.

**archive copy group**

A policy object containing attributes that control the generation, destination, and expiration of archived files.

**archive-retention grace period**

The number of days that the storage manager retains an archived file when the server is unable to rebind the file to an appropriate management class. See also *bind*.

**association**

(1) The defined 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.

(2) On a configuration manager, the defined relationship between a profile and an object such as a policy domain. Profile associations define the configuration information that is distributed to a managed server when it subscribes to the profile.

**audit** To check for logical inconsistencies between information that the server has and the actual condition of the system. The storage manager can audit information about items such as volumes, libraries, and licenses. For example, when a storage manager audits a volume, the server checks for inconsistencies between information about backed-up or archived files that are stored in the database and the actual data that are associated with each backup version or archive copy in server storage.

**authentication**

The process of checking a user's password before permitting user access to the Tivoli Storage Manager server. Authentication can be turned on or off by an administrator with system privilege.

**authentication rule**

A specification that another user can use to either restore or retrieve files from storage.

**authority**

The right to access objects, resources, or functions. See also *privilege class*.

**authorization rule**

A specification that permits another user to either restore or retrieve a user's files from storage.

**authorized user**

A user who has administrative authority for the Tivoli Storage Manager client on a workstation. This user changes passwords, performs open registrations, and deletes file spaces.

**AutoFS**

See *automounted file system*.

**automatic detection**

A feature that detects, reports, and updates the serial number of a drive or library in the database when the path from the local server is defined.

**automatic migration**

The process that is used to automatically move files from a local file system to storage, based on options and settings that are chosen by a root user on a workstation. See also *threshold migration* and *demand migration*.

**automatic reconciliation**

The process that is used to reconcile file systems at regular intervals. The intervals are set by a user with root user authority. See also *reconciliation*.

**automounted file system (AutoFS)**

A file system that is managed by an automounter daemon. The automounter daemon monitors a specified directory path, and automatically mounts the file system to access data.

**B****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 group**

A policy object containing attributes that control the generation, destination, and expiration of backup versions of files. A backup copy group belongs to a management class.

**backup-retention grace period**

The number of days the storage manager retains a backup version after the server is unable to rebind the file to an appropriate management class.

**backup set**

A portable, consolidated group of active versions of backup files that are generated for a backup-archive client.

**backup set collection**

A group of backup sets that are created at the same time and which have the same backup set name, volume names, description, and device classes. The server identifies each backup set in the collection by its node name, backup set name, and file type.

**backup version**

A file or directory that a client node backed up to server storage. More than one backup version can exist in server storage, but only one backup version is the active version. See also *active version* and *inactive version*.

**bindery**

A database that consists of three system files for a NetWare server. The files contain user IDs and user restrictions.

**bind**

To associate a file with a management class name. See *rebind*.

**C****cache**

To place a duplicate copy of a file on random access media when the server migrates a file to another storage pool in the hierarchy.

**cache file**

A snapshot of a logical volume created by Logical Volume Snapshot Agent. Blocks are saved immediately before they are modified during the image backup and their logical extents are saved in the cache files.

**CAD**

See *client acceptor*.

**central scheduler**

A function that permits an administrator to schedule client operations and administrative commands. The operations can be scheduled to occur periodically or on a specific date. See *client schedule* and *administrative command schedule*.

**client** A software program or computer that requests services from a server.

**client acceptor daemon (CAD)**  
See *client acceptor*.

**client acceptor**  
An HTTP service that serves the Java™ applet for the Web client to Web browsers. On Windows systems, the client acceptor is installed and run as a service. On AIX, UNIX, and Linux systems, the client acceptor is run as a daemon, and is also called the *client acceptor daemon* (CAD).

**client domain**  
The set of drives, file systems, or volumes that the user selects to back up or archive data, using the backup-archive client.

**client node**  
A file server or workstation on which the backup-archive client program has been installed, and which has been registered to the server.

**client node session**  
A session in which a client node communicates with a server to perform backup, restore, archive, retrieve, migrate, or recall requests. Contrast with *administrative session*.

**client options file**  
An editable file that identifies the server and communication method, and provides the configuration for backup, archive, hierarchical storage management, and scheduling.

**client option set**  
A group of options that are defined on the server and used on client nodes in conjunction with client options files.

**client-polling scheduling mode**  
A method of operation in which the client queries the server for work. Contrast with *server-prompted scheduling mode*.

**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**  
Pertaining to the model of interaction in distributed data processing in which a program on one computer sends a request to a program on another computer and awaits a response. The requesting program is called a client; the answering program is called a server.

**client system-options file**  
A file, used on AIX, UNIX, or Linux system clients, containing a set of processing options that identify the servers to be contacted for services. This file also specifies communication methods and options for backup, archive, hierarchical storage management, and scheduling. This file is also called the *dsm.sys* file. See also *client user-options file*.

**client user-options file**  
A file that contains the set of processing options that the clients on the system use. The set can include options that determine the server that the client contacts, and options that affect backup operations, archive operations, hierarchical storage management operations, and scheduled operations. This file is also called the *dsm.opt* file. For AIX, UNIX, or Linux systems, see also *client system-options file*.

**closed registration**  
A registration process in which only an administrator can register workstations as client nodes with the server. Contrast with *open registration*.

**collocation**  
The process of keeping all data belonging to a single-client file space, a single client node, or a group of client nodes on a minimal number of sequential-access volumes within a storage pool. Collocation can reduce the number of volumes that must be accessed when a large amount of data must be restored.

**collocation group**  
A user-defined group of client nodes whose data is stored on a minimal number of volumes through the process of collocation.

**commit point**  
A point in time when data is considered consistent.



**communication method**

The method by which a client and server exchange information. See also *Transmission Control Protocol/Internet Protocol*.

**Common Programming Interface for Communications (CPI-C)**

A call-level interface that provides a consistent application programming interface (API) for applications that use program-to-program communications. CPI-C uses LU 6.2 architecture to create a set of interprogram services that can establish and end a conversation, send and receive data, exchange control information, and notify a partner program of errors.

**communication protocol**

A set of defined interfaces that permit computers to communicate with each other.

**compression**

A function that removes repetitive characters, spaces, or strings of characters from the data being processed and replaces the repetitive characters with control characters. Compression reduces the amount of storage space that is required for the data.

**configuration manager**

A server that distributes configuration information, such as policies and schedules, to managed servers according to their profiles. Configuration information can include policy and schedules. See also *managed server* and *profile*.

**conversation**

A connection between two programs over a session that allows them to communicate with each other while processing a transaction.

**copy backup**

A full backup in which the transaction log files are not deleted so that backup procedures that use incremental or differential backups are not disrupted

**copy group**

A policy object containing attributes that control how backup versions or archive copies are generated, where backup versions or archive copies are initially

located, and when backup versions or archive copies expire. A copy group belongs to a management class. See also *archive copy group*, *backup copy group*, *backup version*, and *management class*.

**copy storage pool**

A named set of volumes that contain copies of files that reside in primary storage pools. Copy storage pools are used only to back up the data that is 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 also *primary storage pool* and *destination*.

**CPI-C** See *Common Programming Interface for Communications*.

**D****daemon**

A program that runs unattended to perform continuous or periodic functions, such as network control.

**damaged file**

A physical file in which Tivoli Storage Manager has detected read errors.

**data access control mode**

A mode that controls whether a command can access a migrated file, see a migrated file as zero-length, or receive an input/output error if it attempts to access a migrated file. See also *execution mode*.

**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 number identifies each backup series.

**database snapshot**

A complete backup of the entire database to media that can be taken off-site. When a database snapshot is created, the current database backup series is not interrupted. A database snapshot cannot have incremental database backups associated with it. See also *database backup series*. Contrast with *full backup*.

**data deduplication**

A method of reducing storage needs by eliminating redundant data. Only one instance of the data is retained on storage

media, such as disk or tape. Other instances of the same data are replaced with a pointer to the retained instance.

**data manager server**

A server that collects metadata information for client inventory and manages transactions for the storage agent over the local area network. The data manager server informs the storage agent with applicable library attributes and the target volume identifier.

**data mover**

A device that moves data on behalf of the server. A network-attached storage (NAS) file server is a data mover.

**data storage-management application-program interface (DSMAPI)**

A set of functions and semantics that can monitor events on files, and manage and maintain the data in a file. In an HSM environment, a DSMAPI uses events to notify data management applications about operations on files, stores arbitrary attribute information with a file, supports managed regions in a file, and uses DSMAPI access rights to control access to a file object.

**default management class**

A management class that is assigned to a policy set. This class is used to govern backed up or archived files when a file is not explicitly associated with a specific management class through the include-exclude list.

**deduplication**

See *data deduplication*.

**demand migration**

The process that is used to respond to an out-of-space condition on a file system for which hierarchical storage management (HSM) is active. Files are migrated to server storage until space usage drops to the low threshold that was set for the file system. If the high threshold and low threshold are the same, one file is migrated.

**desktop client**

The group of backup-archive clients that includes clients on Microsoft Windows, Apple, and Novell NetWare operating systems.

**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.

**device class**

A named set of characteristics that are applied to a 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**

(1) For a server, a file that contains information about defined device classes, and, on some servers, defined libraries and drives. The information is a copy of the device configuration information in the database.

(2) For a storage agent, a file that contains the name and password of the storage agent, and information about the server that is managing the SAN-attached libraries and drives that the storage agent uses.

**device driver**

A program that provides an interface between a specific device and the application program that uses the device.

**disaster recovery manager (DRM)**

A function that assists in preparing and using a disaster recovery plan file for the server.

**disaster recovery plan**

A file that is created by the disaster recovery manager (DRM) that contains information about how to recover computer systems if a disaster occurs and scripts that can be run to perform some recovery tasks. The file includes information about the software and hardware that is used by the server, and the location of recovery media.

**domain**

A grouping of client nodes with one or more policy sets, which manage data or storage resources for the client nodes. See *policy domain* or *client domain*.

**DRM** See *disaster recovery manager*.

**DSMAPI**

See *data storage-management application-program interface*.



**dynamic serialization**

A type of copy serialization in which a file or folder is backed up or archived on the first attempt regardless of whether it changes during a backup or archive.

**E**

**EA** See *extended attribute*.

**EB** See *exabyte*.

**EFS** See *Encrypted File System*.

**Encrypted File System (EFS)**

A file system that uses file system-level encryption.

**enterprise configuration**

A method of setting up servers so that the administrator can distribute the configuration of one of the servers to the other servers, using server-to-server communication. See also *configuration manager*, *managed server*, *profile*, and *subscription*.

**enterprise logging**

The process of sending events from a Tivoli Storage Manager server to a designated event server. The event server routes the events to designated receivers, such as to a user exit. See also *event*.

**error log**

A data set or file that is used to record error information about a product or system.

**estimated capacity**

The available space, in megabytes, of a storage pool.

- event** (1) An administrative command or a client operation that is scheduled to be run using Tivoli Storage Manager scheduling.
- (2) A message that an Tivoli Storage Manager server or client issues. Messages can be logged using Tivoli Storage Manager event logging.

**event record**

A database record that describes actual status and results for events.

**event server**

A server to which other servers can send events for logging. The event server routes the events to any receivers that are enabled for the sending server's events.

**exabyte (EB)**

For processor storage, real and virtual storage, and channel volume, 1 152 921 504 606 846 976 bytes. For disk storage capacity and communications volume, 1 000 000 000 000 000 000 bytes.

**exclude**

The process of identifying files in an include-exclude list. This process prevents the files from being backed up or migrated whenever a user or schedule enters an incremental or selective backup operation. A file can be excluded from backup and space management, backup only, or space management only.

**exclude-include list**

See *include-exclude list*.

**execution mode**

A mode that controls the space-management related behavior of commands that run under the **dsmmode** command.

**expiration**

The process by which files, data sets, or objects are identified for deletion because their expiration date or retention period has passed.

**expiring file**

A migrated or premigrated file that has been marked for expiration and removal from storage. If a stub file or an original copy of a premigrated file is deleted from a local file system, or if the original copy of a premigrated file is updated, the corresponding migrated or premigrated file is marked for expiration the next time reconciliation is run.

**extend**

To increase the portion of available space that can be used to store database or recovery log information.

**extended attribute (EA)**

Names or value pairs that are associated with files or directories. There are three classes of extended attributes: user attributes, system attributes, and trusted attributes.

**external library**

A type of library that is provided by Tivoli Storage Manager that permits LAN-free data movement for StorageTek libraries that are managed by Automated

Cartridge System Library Software (ACSLs). To activate this function, the Tivoli Storage Manager library type must be EXTERNAL.

## F

### file access time

On AIX, UNIX, or Linux systems, the time when the file was last accessed.

### file age

For migration prioritization purposes, the number of days since a file was last accessed.

### file device type

A device type that specifies the use of sequential access files on disk storage as volumes.

### file server

A dedicated computer and its peripheral storage devices that are connected to a local area network that stores programs and files that are shared by users on the network.

### file space

A logical space in server storage that contains a group of files that have been backed up or archived by a client node, from a single logical partition, file system, or virtual mount point. Client nodes can restore, retrieve, or delete their file spaces from server storage. In server storage, files belonging to a single file space are not necessarily stored together.

### file space ID (FSID)

A unique numeric identifier that the server assigns to a file space when it is stored in server storage.

### file state

The space management mode of a file that resides in a file system to which space management has been added. A file can be in one of three states: resident, premigrated, or migrated. See also *resident file*, *premigrated file*, and *migrated file*.

### file system migrator (FSM)

A kernel extension that intercepts all file system operations and provides any space management support that is required. If no space management support is required, the operation is passed to the operating system, which performs its normal functions. The file system

migrator is mounted over a file system when space management is added to the file system.

### file system state

The storage management mode of a file system that resides on a workstation on which the hierarchical storage management (HSM) client is installed. A file system can be in one of these states: native, active, inactive, or global inactive.

### frequency

A copy group attribute that specifies the minimum interval, in days, between incremental backups.

**FSID** See *file space ID*.

**FSM** See *file system migrator*.

### full backup

The process of backing up the entire server database. A full backup begins a new database backup series. See also *database backup series* and *incremental backup*. Contrast with *database snapshot*.

### fuzzy backup

A backup version of a file that might not accurately reflect what is currently in the file because the file was backed up at the same time as it was being modified.

### fuzzy copy

A backup version or archive copy of a file that might not accurately reflect the original contents of the file because it was backed up or archived the file while the file was being modified. See also *backup version* and *archive copy*.

## G

### General Parallel File System™

A high-performance shared-disk file system that can provide data access from nodes in a cluster environment.

### gigabyte (GB)

In decimal notation, 1 073 741 824 when referring to memory capacity; in all other cases, it is defined as 1 000 000 000.

### global inactive state

The state of all file systems to which space management has been added when space management is globally deactivated for a client node. When space management is globally deactivated, hierarchical storage management (HSM)

cannot perform migration, recall, or reconciliation. However, a root user can update space management settings and add space management to additional file systems. Users can access resident and premigrated files.

**Globally Unique Identifier (GUID)**

An algorithmically determined number that uniquely identifies an entity within a system.

**GPFS™**

See *General Parallel File System*.

**GPFS node set**

A mounted, defined group of GPFS file systems.

**group backup**

The backup of a group containing a list of files from one or more file space origins.

**GUID** See *Globally Unique Identifier*.

**H**

**hierarchical storage management (HSM)**

A function that automatically distributes and manages data on disk, tape, or both by regarding devices of these types and potentially others as levels in a storage hierarchy that range from fast, expensive devices to slower, cheaper, and possibly removable devices. The objectives are to minimize access time to data and maximize available media capacity.

**hierarchical storage management (HSM) client**

A client program that works with the Tivoli Storage Manager server to provide hierarchical storage management (HSM) for a system. See also *hierarchical storage management* and *space manager client*.

**HSM** See *hierarchical storage management*.

**HSM client**

See *hierarchical storage management client*.

**I**

**ILM** See *information lifecycle management*.

**image** A file system or raw logical volume that is backed up as a single object.

**image backup**

A backup of a full file system or raw logical volume as a single object.

**inactive file system**

A file system for which space

management has been deactivated. Contrast with *active file system*.

**inactive version**

A backup version of a file that is either not the most recent backup version, or that is a backup version of a file that no longer exists on the client system. Inactive backup versions are eligible for expiration processing according to the management class assigned to the file. Contrast with *active version*.

**include-exclude file**

A file containing statements to determine the files to back up and the associated management classes to use for backup or archive. See also *include-exclude list*.

**include-exclude list**

A list of options that include or exclude selected files for backup. An exclude option identifies files that should not be backed up. An include option identifies files that are exempt from the exclusion rules or assigns a management class to a file or a group of files for backup or archive services.

**incremental backup**

(1) A copy of all database data that has changed since the most recent successful full backup operation. An incremental backup is also known as a *cumulative backup image* because each incremental backup includes the contents of the previous incremental backup.

(2) The process of backing up information in the database that is new or changed since the last full backup. Contrast with *full backup*. See also *database backup series*.

(3) For Data Protection for Microsoft Exchange Server, a backup in which the transaction logs are backed up and then cleared.

**individual mailbox restore**

See *mailbox restore*.

**information lifecycle management (ILM)**

GPFS policy-based file management for storage pools and file sets.

**i-node** The internal structure that describes the individual files on AIX, UNIX, or Linux systems. An i-node contains the node, type, owner, and location of a file.

**i-node number**

A number specifying a particular i-node file in the file system.

**IP address**

A unique address for a device or logical unit on a network that uses the IP standard.

**J****job file**

A generated file that contains configuration information for a migration job. The file is XML format and can be created and edited in the hierarchical storage management (HSM) client for Windows client graphical user interface.

**journal-based backup**

A method for backing up Windows clients and AIX clients that exploits the change notification mechanism in a file to improve incremental backup performance by reducing the need to fully scan the file system.

**journal daemon**

On AIX, UNIX, or Linux systems, a program that tracks change activity for files residing in file systems.

**journal service**

In Microsoft Windows, a program that tracks change activity for files residing in file systems.

**K****kilobyte (KB)**

For processor storage, real and virtual storage, and channel volume, 210 or 1 024 bytes. For disk storage capacity and communications volume, 1 000 bytes.

**L**

**LAN** See *local area network*.

**LAN-free data movement**

The movement of client data between a client system and a storage device on a storage area network (SAN), bypassing the local area network. This process is also referred to as *LAN-free data transfer*.

**LAN-free data transfer**

See *LAN-free data movement*.

**leader data**

Bytes of data, from the beginning of a migrated file, that are stored in the file's

corresponding stub file on the local file system. The amount of leader data that is stored in a stub file depends on the stub size that is specified.

**library**

(1) A repository for demountable recorded media, such as magnetic disks and magnetic tapes.

(2) A collection of one or more drives, and possibly robotic devices (depending on the library type), which can be used to access storage volumes.

**library client**

A server that uses server-to-server communication to access a library that is managed by another storage management server. See also *library manager*.

**library manager**

A server that controls device operations when multiple storage management servers share a storage device. See also *library client*.

**local** Pertaining to a device, file, or system that is accessed directly from a user's system, without the use of a communication line.

**local area network (LAN)**

A network that connects several devices in a limited area (such as a single building or campus) and that can be connected to a larger network.

**local shadow volumes**

Data that is stored on shadow volumes localized to a disk storage subsystem.

**LOFS** See *loopback virtual file system*.

**logical file**

A file that is stored in one or more server storage pools, either by itself or as part of an aggregate. See also *aggregate* and *physical file*.

**logical occupancy**

The space that is used by logical files in a storage pool. This space does not include the unused space created when logical files are deleted from aggregate files, so it might be less than the physical occupancy.

**logical unit (LU)**

An access point through which a user or application program accesses the Systems

Network Architecture (SNA) network to communicate with another user or application program.

**logical unit number (LUN)**  
In the Small Computer System Interface (SCSI) standard, a unique identifier that is used to differentiate devices, each of which is a logical unit (LU).

**logical volume**  
A portion of a physical volume that contains a file system.

**logical volume backup**  
A back up of a file system or logical volume as a single object.

**Logical Volume Snapshot Agent (LVSA)**  
Software that can act as the snapshot provider for creating a snapshot of a logical volume during an online image backup.

**loopback virtual file system (LOFS)**  
A file system that is created by mounting a directory over another local directory, also known as mount-over-mount. A LOFS can also be generated using an automounter.

**LU** See *logical unit*.

**LUN** See *logical unit number*.

**LVSA** See *Logical Volume Snapshot Agent*.

**M**

**MB** See *megabyte*.

**macro file**  
A file that contains one or more storage manager administrative commands, which can be run only from an administrative client using the MACRO command. Contrast with *Tivoli Storage Manager command script*.

**mailbox restore**  
A function that restores Microsoft Exchange Server data (from IBM Data Protection for Exchange backups) at the mailbox level or mailbox-item level.

**managed object**  
In Tivoli Storage Manager, a definition in the database of a managed server that was distributed to the managed server by a configuration manager. When a managed server subscribes to a profile, all objects that are associated with that

profile become managed objects in the database of the managed server. In general, a managed object cannot be modified locally on the managed server. Objects can include policy, schedules, client option sets, server scripts, administrator registrations, and server and server group definitions.

#### **managed server**

A Tivoli Storage Manager server that receives configuration information from a configuration manager using a subscription to one or more profiles. Configuration information can include definitions of objects such as policy and schedules. See also *configuration manager*, *subscription*, and *profile*.

#### **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. See also *copy group*, *space manager client*, *bind*, and *rebind*.

#### **maximum transmission unit**

The largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the maximum transmission unit for Ethernet is 1500 bytes.

#### **megabyte (MB)**

- (1) 1 048 576 bytes (two to the twentieth power) when used in this publication.
- (2) For processor storage, real and virtual storage, and channel volume, 2 to the power of 20 or 1 048 576 bits. For disk storage capacity and communications volume, 1 000 000 bits.

#### **metadata**

Data that describes the characteristics of data; descriptive data.

#### **migrate**

To move data from one storage location to another. In Tivoli Storage Manager products, migrating can mean moving data from a client node to server storage, or moving data from one storage pool to the next storage pool defined in the server storage hierarchy. In both cases the



movement is controlled by policy, such as thresholds that are set. See also *migration threshold*.

**migrated file**

A file that has been copied from a local file system to Tivoli Storage Manager storage. For HSM clients on UNIX or Linux systems, the file is replaced with a stub file on the local file system. On Windows systems, creation of the stub file is optional. See also *stub file* and *resident file*. For HSM clients on UNIX or Linux systems, contrast with *premigrated file*.

**migrate-on-close recall mode**

A mode that causes a migrated file to be recalled back to its originating file system temporarily. Contrast with *normal recall mode* and *read-without-recall recall mode*.

**migration job**

A specification of files to migrate, and actions to perform on the original files after migration. See also *job file*.

**migration threshold**

High and low capacities for storage pools or file systems, expressed as percentages, at which migration is set to start and stop.

**mirroring**

The process of writing the same data to multiple locations at the same time. Mirroring data protects against data loss within the recovery log.

**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 mode* and *absolute mode*.

**modified mode**

In storage management, a backup copy-group mode that specifies that a file is considered for incremental backup only if it has changed since the last backup. A file is considered a changed file if the date, size, owner, or permissions of the file have changed. See also *absolute mode*.

**mount limit**

The maximum number of volumes that can be simultaneously accessed from the same device class. The mount limit determines the maximum number of mount points. See also *mount point*.

**mount point**

On the Tivoli Storage Manager server, a logical drive through which volumes in a sequential access device class are accessed. For removable-media device types, such as tape, a mount point is a logical drive that is associated with a physical drive. For the file device type, a mount point is a logical drive that is associated with an I/O stream. The number of mount points for a device class is defined by the value of the mount limit attribute for that device class. See also *mount limit*.

**mount retention period**

The maximum number of 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**

The maximum number of minutes that the server waits for a sequential-access volume mount request to be satisfied before canceling the request.

**MTU** See *maximum transmission unit*.

**N****Nagle algorithm**

An algorithm that reduces congestion of TCP/IP networks by combining smaller packets and sending them together.

**named pipe**

A type of interprocess communication that permits message data streams to pass between peer processes, such as between a client and a server.

**NAS** See *network-attached storage*.

**NAS node**

A client node that is a network-attached storage (NAS) file server. Data for the NAS node is transferred by a NAS file server that is controlled by the network data management protocol (NDMP). A NAS node is also called a NAS file server node.

**native file system**

A file system that is locally added to the file server and is not added for space management. The hierarchical storage

- manager (HSM) client does not provide space management services to the file system.
- native format**  
A format of data that is written to a storage pool directly by the Tivoli Storage Manager server. Contrast with *non-native data format*.
- NDMP**  
See *Network Data Management Protocol*.
- NetBIOS**  
See *Network Basic Input/Output System*.
- network-attached storage (NAS) file server**  
A dedicated storage device with an operating system that is optimized for file-serving functions. A NAS file server can have the characteristics of both a node and a data mover.
- Network Basic Input/Output System (NetBIOS)**  
A standard interface to networks and personal computers that is used on local area networks to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not have to handle the details of LAN data link control (DLC) protocols.
- Network Data Management Protocol (NDMP)**  
A protocol that allows a network storage-management application to control the backup and recovery of an NDMP-compliant file server, without installing vendor-acquired software on that file server.
- network data-transfer rate**  
A rate that is calculated by dividing the total number of bytes that are transferred by the data transfer time. For example, this rate can be the time that is spent transferring data over a network.
- node**  
A file server or workstation on which the backup-archive client program has been installed, and which has been registered to the server.
- node name**  
A unique name that is used to identify a workstation, file server, or PC to the server.
- node privilege class**  
A privilege class that gives an administrator the authority to remotely access backup-archive clients for a specific client node or for all clients in a policy domain. See also *privilege class*.
- non-native data format**  
A format of data that is written to a storage pool that differs from the format that the server uses for operations.
- normal recall mode**  
A mode that causes a migrated file to be copied back to its originating file system when it is accessed.
- O**
- offline volume backup**  
A backup in which the volume is locked so that no other system applications can access it during the backup operation.
- online volume backup**  
A backup in which the volume is available to other system applications during the backup operation.
- open registration**  
A registration process in which users can register their workstations as client nodes with the server. Contrast with *closed registration*.
- operator privilege class**  
A privilege class that gives an administrator the authority to disable or halt the server, enable the server, cancel server processes, and manage removable media. See also *privilege class*.
- options file**  
A file that contains processing options. On Windows and NetWare systems, the file is called dsm.opt. On AIX, UNIX, Linux, and Mac OS X systems, the file is called dsm.sys.
- originating file system**  
The file system from which a file was migrated. When a file is recalled using normal or migrate-on-close recall mode, it is always returned to its originating file system.
- orphaned stub file**  
A file for which no migrated file can be found on the Tivoli Storage Manager server that the client node is contacting for space management services. For example, a stub file can be orphaned when the client system-options file is

modified to contact a server that is different than the one to which the file was migrated.

**out-of-space protection mode**

A mode that controls whether the program intercepts out-of-space conditions. See also *execution mode*.

**P**

**pacing**

In SNA, a technique by which the receiving system controls the rate of transmission of the sending system to prevent overrun.

**packet** In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole.

**page** A defined unit of space on a storage medium or within a database volume.

**partial-file recall mode**

A recall mode that causes the hierarchical storage management (HSM) function to read just a portion of a migrated file from storage, as requested by the application accessing the file.

**password generation**

A process that creates and stores a new password in an encrypted password file when the old password expires. Automatic generation of a password prevents password prompting. Password generation can be set in the options file (passwordaccess option). See also *options file*.

**path** An object that defines a one-to-one relationship between a source and a destination. Using the path, the source accesses the destination. Data can flow from the source to the destination, and back. An example of a source is a data mover (such as a network-attached storage [NAS] file server), and an example of a destination is a tape drive.

**pattern-matching character**

See *wildcard character*.

**physical file**

A file that is stored in one or more storage pools, consisting of either a single logical file, or a group of logical files that are packaged together as an aggregate. See also *aggregate* and *logical file*.

**physical occupancy**

The amount of space that is used by physical files in a storage pool. This space includes the unused space that is created when logical files are deleted from aggregates. See also *physical file*, *logical file*, and *logical occupancy*.

**plug-in**

A self-contained software component that modifies (adds, or changes) the function in a particular system. When a plug-in is added to a system, the foundation of the original system remains intact.

**policy domain**

A grouping of policy users with one or more policy sets, which manage data or storage resources for the users. The users are client nodes that are associated with the policy domain.

**policy privilege class**

A privilege class that gives an administrator the authority to manage policy objects, register client nodes, and schedule client operations for client nodes. Authority can be restricted to certain policy domains. See also *privilege class*.

**policy set**

A group of rules in a policy domain. The rules specify how data or storage resources are automatically managed for client nodes in the policy domain. Rules can be contained in management classes. See also *active policy set* and *management class*.

**premigrated file**

A file that has been copied to Tivoli Storage Manager storage, but has not been replaced with a stub file on the local file system. An identical copy of the file resides both on the local file system and in Tivoli Storage Manager storage. Premigrated files occur on UNIX and Linux file systems to which space management has been added. Contrast with *migrated file* and *resident file*.

**premigrated files database**

A database that contains information about each file that has been premigrated to Tivoli Storage Manager storage. The database is stored in a hidden directory



named .SpaceMan in each file system to which space management has been added.

**premigration**

The process of copying files that are eligible for migration to Tivoli Storage Manager storage, but leaving the original file intact on the local file system.

**premigration percentage**

A space management setting that controls whether the next eligible candidates in a file system are premigrated following threshold or demand migration.

**primary storage pool**

A named set of volumes that the server uses to store backup versions of files, archive copies of files, and files migrated from client nodes. See also *destination* and *copy storage pool*.

**privilege class**

A level of authority that is granted to an administrator. The privilege class determines which administrative tasks the administrator can perform. See also *node privilege class*, *operator privilege class*, *policy privilege class*, *storage privilege class*, and *system privilege class*.

**profile**

A named group of configuration information that can be distributed from a configuration manager when a managed server subscribes. Configuration information can include registered administrator IDs, policies, client schedules, client option sets, administrative schedules, storage manager command scripts, server definitions, and server group definitions. See also *configuration manager* and *managed server*.

**Q**

- quota** (1) For HSM on AIX, UNIX, or Linux systems, the limit (in megabytes) on the amount of data that can be migrated and premigrated from a file system to server storage.
- (2) For HSM on Windows systems, a user-defined limit to the space that is occupied by recalled files.

**R****randomization**

The process of distributing schedule start

times for different clients within a specified percentage of the schedule's startup window.

**raw logical volume**

A portion of a physical volume that is comprised of unallocated blocks and has no journaled file system (JFS) definition. A logical volume is read/write accessible only through low-level I/O functions.

**read-without-recall recall mode**

A mode that causes hierarchical storage management (HSM) to read a migrated file from storage without storing it back on the local file system. The last piece of information read from the file is stored in a buffer in memory on the local file system. Contrast with *normal recall mode* and *migrate-on-close recall mode*.

**rebind**

To associate a backed-up file with a new management class name. For example, rebinding occurs when the management class associated with a file is deleted. See also *bind*.

**recall** In Tivoli Storage Manager, to copy a migrated file from server storage back to its originating file system using the space management client. See also *transparent recall*, *selective recall*, and *recall mode*.

**recall mode**

A mode that is assigned to a migrated file with the *dsmattr* command that determines how the file is processed when it is recalled. It determines whether the file is stored on the local file system, is migrated back to Tivoli Storage Manager storage when it is closed, or is read from Tivoli Storage Manager storage without storing it on the local file system.

**receiver**

A server repository that contains a log of server and client messages as events. For example, a receiver can be a file exit, a user exit, or the Tivoli Storage Manager server console and activity log. See also *event*.

**reclamation**

The process of consolidating the remaining data from many sequential-access volumes onto fewer, new sequential-access volumes.

**reclamation threshold**

The percentage of space that a sequential-access media volume must have before the server can reclaim the volume. Space becomes reclaimable when files are expired or are deleted.

**reconciliation**

The process of synchronizing a file system with the Tivoli Storage Manager server, and then removing old and obsolete objects from the Tivoli Storage Manager server.

**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. The recovery log consists of the active log (including the log mirror) and archive logs.

**register**

To define a client node or administrator ID that can access the server.

**registry**

A repository that contains access and configuration information for users, systems, and software.

**resident file**

On a Windows system, a complete file on a local file system that might also be a migrated file because a migrated copy can exist in Tivoli Storage Manager storage. On a UNIX or Linux system, a complete file on a local file system that has not been migrated or premigrated, or that has been recalled from Tivoli Storage Manager storage and modified. Contrast with *stub file* and *premigrated file*. See *migrated file*.

**restore**

To copy information from its backup location to the active storage location for use. For example, to copy information from server storage to a client workstation.

**retention**

The amount of time, in days, that inactive backed-up or archived files are kept in the storage pool before they are deleted. Copy group attributes and default retention grace periods for the domain define retention.

**retrieve**

To copy archived information from the

storage pool to the workstation for use. The retrieve operation does not affect the archive version in the storage pool.

**roll back**

To remove changes that were made to database files since the last commit point.

**root user**

A system user who operates without restrictions. A root user has the special rights and privileges needed to perform administrative tasks.

**S**

**SAN** See *storage area network*.

**schedule**

A database record that describes client operations or administrative commands to be processed. See *administrative command schedule* and *client schedule*.

**scheduling mode**

The type of scheduling operation for the server and client node that supports two scheduling modes: client-polling and server-prompted.

**scratch volume**

A labeled volume that is either blank or contains no valid data, that is not defined, and that is available for use.

**script** A series of commands, combined in a file, that carry out a particular function when the file is run. Scripts are interpreted as they are run. Contrast with *Tivoli Storage Manager command script*.

**Secure Sockets Layer (SSL)**

A security protocol that provides communication privacy. With SSL, client/server applications can communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

**selective backup**

The process of backing up certain files or directories from a client domain. The files that are backed up are those that are not excluded in the include-exclude list. The files must meet the requirement for serialization in the backup copy group of the management class that is assigned to each file. Contrast with *incremental backup*.

**selective migration**

The process of copying user-selected files

- from a local file system to Tivoli Storage Manager storage and replacing the files with stub files on the local file system. Contrast with *threshold migration* and *demand migration*.
- selective recall**  
The process of copying user-selected files from Tivoli Storage Manager storage to a local file system. Contrast with *transparent recall*.
- serialization**  
The process of handling files that are modified during backup or archive processing. See *dynamic serialization*, *static serialization*, *shared static serialization*, and *shared dynamic serialization*.
- server** A software program or a computer that provides services to other software programs or other computers.
- server options file**  
A file that contains settings that control various server operations. These settings affect such things as communications, devices, and performance.
- server-prompted scheduling mode**  
A client/server communication technique where the server contacts the client node when tasks must be done. Contrast with *client-polling scheduling mode*.
- server storage**  
The primary, copy, and active-data storage pools that are used by the server to store user files such as backup versions, archive copies, and files migrated from space manager client nodes (space-managed files). See also *active-data pool*, *primary storage pool*, *copy storage pool*, *storage pool volume*, and *volume*.
- session**  
A logical or virtual connection between two stations, software programs, or devices on a network that allows the two elements to communicate and exchange data.
- session resource usage**  
The amount of wait time, processor time, and space that is used or retrieved during a client session.
- shared dynamic serialization**  
A value for serialization that specifies that a file must not be backed up or archived if it is being modified during the operation. Tivoli Storage Manager retries the backup or archive operation a number of times; if the file is being modified during each attempt, Tivoli Storage Manager will back up or archive the file on its last try. See also *serialization*. Contrast with *dynamic serialization*, *shared static serialization*, and *static serialization*.
- shared library**  
A library device that is used by multiple storage manager servers.
- shared static serialization**  
A copy-group serialization value that specifies that a file must not be modified during a backup or archive operation. Tivoli Storage Manager attempts to retry the operation a number of times. If the file is in use during each attempt, the file is not backed up or archived. See also *serialization*. Contrast with *dynamic serialization*, *shared dynamic serialization*, and *static serialization*.
- snapshot**  
An image backup type that consists of a point-in-time view of a volume.
- space-managed file**  
A file that is migrated from a client node by the space manager client. The space manager client recalls the file to the client node on demand.
- space management**  
The process of keeping sufficient free storage space available on a local file system for new data by migrating files to server storage. Synonymous with *hierarchical storage management*.
- space manager client**  
A program that runs on a UNIX or Linux system to manage free space on the local file system by migrating files to server storage. The program can recall the files either automatically or selectively. Also called *hierarchical storage management (HSM) client*.
- space monitor daemon**  
A daemon that checks space usage on all file systems for which space management is active, and automatically starts threshold migration when space usage on a file system equals or exceeds its high threshold.

**sparse file**

A file that is created with a length greater than the data it contains, leaving empty spaces for the future addition of data.

**special file**

On AIX, UNIX, or Linux systems, a file that defines devices for the system, or temporary files that are created by processes. There are three basic types of special files: first-in, first-out (FIFO); block; and character.

**SSL** See *Secure Sockets Layer*.

**stabilized file space**

A file space that exists on the server but not on the client.

**stanza** A group of lines in a file that together have a common function or define a part of the system. Each stanza is identified by a name that occurs in the first line of the stanza. Depending on the type of file, a stanza is ended by the next occurrence of a stanza name in the file, or by an explicit end-of-stanza marker. A stanza can also be ended by the end of the file.

**startup window**

A time period during which a schedule must be initiated.

**static serialization**

A copy-group serialization value that specifies that a file must not be modified during a backup or archive operation. If the file is in use during the first attempt, the storage manager cannot back up or archive the file. See also *serialization*. Contrast with *dynamic serialization*, *shared dynamic serialization*, and *shared static serialization*.

**storage agent**

A program that enables the backup and restoration of client data directly to and from storage attached to a storage area network (SAN).

**storage area network (SAN)**

A dedicated storage network that is tailored to a specific environment, combining servers, systems, storage products, networking products, software, and services.

**storage hierarchy**

(1) A logical order of primary storage pools, as defined by an administrator. The

order is typically based on the speed and capacity of the devices that the storage pools use. The storage hierarchy is defined by identifying the next storage pool in a storage pool definition. See also *storage pool*.

(2) An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include: main storage, such as memory and direct-access storage device (DASD) cache; primary storage (DASD containing user-accessible data); migration level 1 (DASD containing data in a space-saving format); and migration level 2 (tape cartridges containing data in a space-saving format).

**storage pool**

A named set of storage volumes that are the destination that is used to store client data. A storage pool contains backup versions, archive copies, and files that are migrated from space manager client nodes. A primary storage pool is backed up to a copy storage pool. See also *primary storage pool*, *copy storage pool*, and *active-data pool*.

**storage pool volume**

A volume that has been assigned to a storage pool. See also *volume*, *active-data pool*, *copy storage pool*, and *primary storage pool*.

**storage privilege class**

A privilege class that gives an administrator the authority to control how storage resources for the server are allocated and used, such as monitoring the database, the recovery log, and server storage. See also *privilege class*.

**stub**

A shortcut on the Windows file system that is generated by the hierarchical storage management (HSM) client for a migrated file that allows transparent user access. A stub is the sparse file representation of a migrated file, with a reparse point attached.

**stub file**

A file that replaces the original file on a local file system when the file is migrated to storage. A stub file contains the information that is necessary to recall a migrated file from Tivoli Storage Manager storage. It also contains additional

information that can be used to eliminate the need to recall a migrated file.

**stub file size**

The size of a file that replaces the original file on a local file system when the file is migrated to Tivoli Storage Manager storage. The size that is specified for stub files determines how much leader data can be stored in the stub file. The default for stub file size is the block size defined for a file system minus 1 byte.

**subscription**

In a Tivoli environment, the process of identifying the subscribers that the profiles are distributed to. For Tivoli Storage Manager, a subscription is the process by which a managed server receives configuration information associated with a particular profile on a configuration manager. See also *managed server*, *configuration manager*, and *profile*.

**Systems Network Architecture (SNA)**

The description of the logical structure, formats, protocols, and operational sequences for transmitting information through and controlling the configuration and operation of networks.

**system privilege class**

A privilege class that gives an administrator the authority to issue all server commands. See also *privilege class*.

**T**

**tape library**

A set of equipment and facilities that support an installation's tape environment. The tape library can include tape storage racks, mechanisms for automatic tape mounting, a set of tape drives, and a set of related tape volumes mounted on those drives.

**tape volume prefix**

The high-level-qualifier of the file name or the data set name in the standard tape label.

**target node**

A client node for which other client nodes (called agent nodes) have been granted proxy authority. The proxy authority allows the agent nodes to perform operations such as backup and restore on behalf of the target node, which owns the data.

**TCA** See *trusted communications agent*.

**TCP/IP**

See *Transmission Control Protocol/Internet Protocol*.

**threshold migration**

The process of moving files from a local file system to Tivoli Storage Manager storage based on the high and low thresholds that are defined for the file system. Contrast with *demand migration*, *selective migration*, and *migration job*.

**throughput**

In storage management, the total bytes in the workload, excluding overhead, that are backed up or restored, divided by elapsed time.

**timeout**

A time interval that is allotted for an event to occur or complete before operation is interrupted.

**timestamp control mode**

A mode that determines whether commands preserve the access time for a file or set it to the current time.

**Tivoli Storage Manager command script**

A sequence of Tivoli Storage Manager administrative commands that are stored in the database of the Tivoli Storage Manager server. The script can run from any interface to the server. The script can include substitution for command parameters and conditional logic.

**tombstone object**

A small subset of attributes of a deleted object. The tombstone object is retained for a specified period, and at the end of the specified period, the tombstone object is permanently deleted.

**Transmission Control Protocol/Internet Protocol (TCP/IP)**

An industry-standard, nonproprietary set of communication protocols that provides reliable end-to-end connections between applications over interconnected networks of different types.

**transparent recall**

The process that is used to automatically recall a file to a workstation or file server when the file is accessed. See also *recall mode*. Contrast with *selective recall*.



**trusted communications agent (TCA)**

A program that handles the sign-on password protocol when clients use password generation.

**U**

**UCS-2** A 2-byte (16-bit) encoding scheme based on ISO/IEC specification 10646-1. UCS-2 defines three levels of implementation: Level 1-No combining of encoded elements allowed; Level 2-Combining of encoded elements is allowed only for Thai, Indic, Hebrew, and Arabic; Level 3-Any combination of encoded elements are allowed.

**UNC** See *Universal Naming Convention name*.

**Unicode**

A character encoding standard that supports the interchange, processing, and display of text that is written in the common languages around the world, plus some classical and historical texts. The Unicode standard has a 16-bit character set defined by ISO 10646.

**Unicode-enabled file space**

Unicode file space names provide support for multilingual workstations without regard for the current locale.

**Unicode transformation format 8**

Unicode Transformation Format (UTF), 8-bit encoding form, which is designed for ease of use with existing ASCII-based systems. The CCSID value for data in UTF-8 format is 1208.

**Universal Naming Convention (UNC) name**

A name that is used to access a drive or directory containing files shared across a network. The UNC name includes the system name and a SharePoint name that represents the shared drive or directory.

**Universally Unique Identifier (UUID)**

The 128-bit numerical identifier that is used to ensure that two components do not have the same identifier.

**UTF-8** See *Unicode transformation format 8*.

**UUID** See *Universally Unique Identifier*.

**V****validate**

To check a policy set for conditions that can cause problems if that policy set becomes the active policy set. For

example, the validation process checks whether the policy set contains a default management class.

**version**

A backup copy of a file stored in server storage. The most recent backup copy of a file is the active version. Earlier copies of the same file are inactive versions. The number of versions retained by the server is determined by the copy group attributes in the management class.

**virtual file space**

A representation of a directory on a network-attached storage (NAS) file system as a path to that directory.

**virtual volume**

An archive file on a target server that represents a sequential media volume to a source server.

**volume**

A discrete unit of storage on disk, tape or other data recording medium that supports some form of identifier and parameter list, such as a volume label or input/output control. See also *scratch volume*, and *storage pool volume*.

**volume history file**

A file that contains information about volumes that have been used by the server for database backups and for export of administrator, node, policy, or server data. The file also has information about sequential-access storage pool volumes that have been added, reused, or deleted. The information is a copy of volume information that is recorded in the server database.

**Volume Shadow Copy Service**

A set of Microsoft application-programming interfaces (APIs) that you can use to create shadow copy backups of volumes, exact copies of files, including all open files, and so on.

**VSS** See *Volume Shadow Copy Service*.

**VSS Backup**

A backup operation that uses Microsoft Volume Shadow Copy Service (VSS) technology. The backup operation produces an online snapshot (point-in-time consistent copy) of Exchange data. This copy can be stored

on local shadow volumes or on Tivoli Storage Manager server storage.

**VSS Fast Restore**

A function that uses a Microsoft Volume Shadow Copy Service (VSS) software provider to restore VSS Backups (IBM Data Protection for Exchange database files and log files) that reside on local shadow volumes.

**VSS Instant Restore**

A volume-level hardware-assisted Microsoft Volume Shadow Copy Service (VSS) function where target volumes that contain the snapshot are copied back to the original source volumes.

**VSS offloaded backup**

A backup operation that uses a Microsoft Volume Shadow Copy Service (VSS) hardware provider (installed on an alternate system) to move IBM Data Protection for Exchange data to the Tivoli Storage Manager server. This type of backup operation shifts the backup load from the production system to another system.

**VSS Restore**

A function that uses a Microsoft Volume Shadow Copy Service (VSS) software provider to restore VSS Backups (IBM Data Protection for Exchange database files and log files) that reside on Tivoli Storage Manager server storage to their original location.

**W**

**wildcard character**

A special character such as an asterisk (\*) or a question mark (?) that can be used to represent one or more characters. Any character or set of characters can replace the wildcard character.

**workstation**

A configuration of input/output equipment at which an operator works. A workstation is a terminal or microcomputer at which a user can run applications and that is usually connected to a mainframe or a network.

**worldwide name**

A 64-bit, unsigned name identifier that is unique.

**workload partition (WPAR)**

A partition within a single operating system instance.





---

# Index

## Numerics

3490 device type (CARTRIDGE) 162  
3494SHARED option 1175  
3570 device type 140  
3590 device type 143  
3592 device type 147, 994  
4MM device type 152  
8MM device type 156

## A

ACCEPT DATE command 28  
ACCESS parameter 339, 1158  
accessibility features 1279  
account record, setting 900  
ACSACCESSID option 1176  
ACSLCKDRIVE option 1177  
ACSL library 227  
ACSQUICKINIT option 1178  
ACSTIMEOUTX option 1179  
action, restore 114, 264, 1094  
ACTIVATE POLICYSET command 29  
activating policy sets 29  
active log 1261  
active log mirror 1261  
active-data pool  
    defining a new active-data pool 325  
    identifying with a QUERY command 788  
    restoring data  
        restoring storage pools 872  
        restoring volumes 876  
    specifying as backups for primary storage pools 299, 314, 1125, 1139  
    updating an existing active-data pool 1147  
active-data pools  
    list 638  
ACTIVELOGDIR option 1180  
ACTIVELOGSIZE option 1181  
activity log  
    querying 587  
    setting retention period 901  
    setting size limit 901  
administrative client  
    batch mode 1, 3  
    console mode 1, 2  
    continuation characters 11  
    interactive mode 1, 3  
    mount mode 1, 2  
    options 5  
    privilege classes 18  
    starting 2  
    stopping 2  
    using 1  
administrative client options  
    alwaysprompt 5  
    commadelimited 5  
    consolemode 5  
    dataonly 6  
    displaymode 6  
    id 6  
administrative client options (*continued*)  
    itemcommit 6  
    mountmode 6  
    newlineafterprompt 6  
    noconfirm 6  
    outfile 2, 6  
    password 6  
    quiet 6  
    serveraddress 6  
    tabdelimited 7  
    tcpport 7  
    tcpserveraddress 7  
administrative command  
    background processing 14  
    components 8  
    entry rules 8  
    foreground processing 14  
    help 495  
    issued by all administrators 25  
    issued with operator privilege 23  
    issued with policy privilege 21  
    issued with storage privilege 22  
    issued with system privilege 18  
    naming conventions 12  
    privilege class for 18  
    scheduling 272  
    using 27  
    using wildcard characters 13  
administrative commands  
    parameters, entering 10  
administrative schedule  
    associating with profile 248  
    copying 99  
    defining 272  
    deleting 387  
    distributing 254  
    updating 1090  
administrative session, canceling 69  
administrator  
    associating with profile 248  
    distributing 254  
    exporting 429  
    importing 502  
    locking 530  
    querying 592  
    registering 833  
    removing 848  
    renaming 852  
    updating 966  
ADMINONCLIENTPORT option 1182  
    administrative sessions using TCPPORT 1182  
    using with TCPADMINPORT 1240  
affect performance, raw partitions 292  
AIXASYNCIO 1183  
ALIASHALT server option 1184  
ANR9999D error message 909  
ANS1082E message 116, 266, 1095  
ANS1102E message 116, 266, 1095  
ARCHDELETE parameter 839, 1073  
ARCHFAILOVERLOGDIR option 1185

- archive
  - defining 132
  - description of 132
  - frequency 133
  - retention grace period 214, 1053
- archive copy group
  - attributes of 132
  - defining 132
  - serialization 132
- archive file deletion by a client 839, 1073
- archive log failover directory 1261
- ARCHIVELOGDIR option 1186
- ASSIGN DEFMGMTCLASS command 31
- assigning
  - client node to domain 837, 1071
  - default management class 31
- ASSISTVCRRECOVERY option 1187
- association of client with schedule
  - defining 106
  - deleting 344
  - querying 596
- association, object with profile
  - defining 248
  - deleting 380
- AUDIT LIBRARY command 34
- AUDIT LICENSES command 36
- audit occupancy
  - displaying storage pool utilization 598
  - querying 598
- AUDIT VOLUME command 37
- auditing
  - library 34
  - storage pool volume 37
- AUDITSTORAGE option 1188
- authentication
  - password 905
  - setting open or closed 905
- authority
  - client access 488
  - client owner 488
  - for command use 18
  - granting 488
  - QUERYAUTH option 1219
  - REQSYSAUTHOUTFILE option 1223
  - revoking 881
  - unlocking 962
- AUTOMIGNONUSE parameter 236, 1069

## B

- BACKDELETE parameter 840, 1073
- background process
  - canceled 65
  - query process 65, 736
- backup
  - backup file deletion 840, 1073
  - client 127
  - database 44
  - device information 47
  - primary storage pool 54
  - scheduling for client 261
  - volume history 58
- backup copy group
  - attributes of 127
  - defining 127
  - serialization 127
  - TOCDestination parameter 130

- backup copy group (*continued*)
  - updating 977
- BACKUP DB command 44
- BACKUP DEVCONFIG command 47
- BACKUP NODE command 49
- backup set
  - defining 108
  - deleting 346
  - displaying contents of 606
  - generating 476
  - node groups
    - defining a group 238
    - defining a member 239
    - deleting a group 375
    - deleting a member 376
    - querying a group 722
    - updating a group 1079
  - querying 601
  - updating retention period 968
- BACKUP STGPOOL command 54
- BACKUP VOLHISTORY command 58
- batch mode
  - restrictions 3
  - starting and ending 3
  - using 3
- BEGIN EVENTLOGGING command 60
- BEGINDATE parameter 588, 662
- BEGINTIME parameter 588, 662, 709

## C

- CACHE parameter
  - DEFINE STGPOOL, primary random access 297
  - UPDATE STGPOOL, primary random access 1123
- cancel
  - administrative session 69
  - client node session 69
  - expiration process 63
  - mount requests 67
  - process 65
  - restartable restore session 68
- CANCEL EXPIRATION command 63
- CANCEL PROCESS 15
- CANCEL PROCESS command 65
- CANCEL REQUEST command 67
- CANCEL RESTORE command 68
- CANCEL SESSION command 69
- canceled commands 15
- CAP parameter
  - MOVE DRMEDIA 77, 544, 559
- CARTRIDGE device type 162
- CD support 205
- Centra storage device
  - authentication and authorization 166, 1010
  - defining a device class 166
  - updating a device class 1010
- changing date and time on the server 28
- characters available for specifying passwords 12
- CHECKIN LIBVOLUME command 70
- checking in
  - library volume 70
  - with labeling 70
- checking out a library volume 77
- CHECKLABEL parameter
  - AUDIT LIBRARY 35
  - CHECKIN LIBVOLUME 74
  - CHECKOUT LIBVOLUME 80

- CHECKOUT LIBVOLUME command 77
- CLASSES parameter
  - GRANT AUTHORITY 488
  - REVOKE AUTHORITY 881
- CLEAN DRIVE command 83
- cleaning a drive, frequency 218
- client access authority 488
- client node session
  - canceling 69
  - scheduling 260
- client option 118
- client owner authority 488
- client-server options, setting 1171
- client, backing up subfiles for 952
- COLLOCATE parameter
  - DEFINE STGPOOL
    - active-data pool 326
    - copy sequential access 318
    - primary sequential access 309
  - UPDATE STGPOOL
    - active-data pool 1148
    - copy sequential access 1142
    - primary sequential access 1134
- collocation
  - group
    - defining 122
    - deleting 353
    - querying 611
    - updating 975
  - group member
    - defining 124
    - deleting 354
  - specifying for a storage pool
    - active-data pool 326, 1148
    - copy sequential access 318, 1142
    - primary sequential access 309, 1134
- command line interface
  - administrative client 1
  - server console 1
  - using 1
- command line, Administration Center 7
- command output, formatting 4
- command routing
  - defining server groups for 16
  - defining servers for 16
- command scripts
  - copying 103
  - defining 281
  - deleting 390
  - querying 765
  - renaming 857
  - running 887
  - running commands in parallel 577
  - running commands serially 897
  - updating 1110
- commands 407
  - ACCEPT DATE 28
  - ACTIVATE POLICYSET 29
  - ASSIGN DEFMGMTCLASS 31
  - AUDIT LIBRARY 34
  - AUDIT LICENSES 36
  - AUDIT VOLUME 37
  - BACKUP DB 44
  - BACKUP DEVCONFIG 47
  - BACKUP NODE 49
  - BACKUP STGPOOL 54
  - BACKUP VOLHISTORY 58

- commands (continued)
  - BEGIN EVENTLOGGING 60
  - CANCEL EXPIRATION 63
  - CANCEL EXPORT 64
  - CANCEL PROCESS 65
  - CANCEL REQUEST 67
  - CANCEL RESTORE 68
  - CANCEL SESSION 69
  - changes in V6 xxix
  - CHECKIN LIBVOLUME 70
  - CHECKOUT LIBVOLUME 77
  - CLEAN DRIVE 83
  - COMMIT 84
  - COPY ACTIVATEDATA 86
  - COPY CLOPTSET 90
  - COPY DOMAIN 91
  - COPY MGMTCLASS 93
  - COPY POLICYSET 95
  - COPY PROFILE 97
  - COPY SCHEDULE 99
  - COPY SCRIPT 103
  - COPY SERVERGROUP 104
  - DEFINE ASSOCIATION 106
  - DEFINE BACKUPSET 108
  - DEFINE CLIENTACTION 113
  - DEFINE CLIENTOPT 118
  - DEFINE CLOPTSET 121
  - DEFINE COLLOGROUP 122
  - DEFINE COLLOCMEMBER 124
  - DEFINE COPYGROUP 126
  - DEFINE DATAMOVER 136
  - DEFINE DEVCLASS 139
  - DEFINE DOMAIN 214
  - DEFINE DRIVE 216
  - DEFINE EVENTSERVER 220
  - DEFINE GRPMEMBER 221
  - DEFINE LIBRARY 222
  - DEFINE MACHINE 231
  - DEFINE MACHNODEASSOCIATION 233
  - DEFINE MGMTCLASS 235
  - DEFINE NODEGROUP 238
  - DEFINE NODEGROUPMEMBER 239
  - DEFINE PATH 240
  - DEFINE POLICYSET 246
  - DEFINE PROFASSOCIATION 248
  - DEFINE PROFILE 254
  - DEFINE RECMEDMACHASSOCIATION 256
  - DEFINE RECOVERYMEDIA 258
  - DEFINE SCHEDULE 260
  - DEFINE SCRIPT 281
  - DEFINE SERVER 283
  - DEFINE SERVERGROUP 288
  - DEFINE SPACETRIGGER 289
  - DEFINE STGPOOL 292
  - DEFINE SUBSCRIPTION 333
  - DEFINE VIRTUALFSMAPPING 335
  - DEFINE VOLUME 337
  - DELETE ASSOCIATION 344
  - DELETE BACKUPSET 346
  - DELETE CLIENTOPT 351
  - DELETE CLOPTSET 352
  - DELETE COLLOGROUP 353
  - DELETE COLLOCMEMBER 354
  - DELETE COPYGROUP 356
  - DELETE DATAMOVER 358
  - DELETE DEVCLASS 359
  - DELETE DOMAIN 360

commands (*continued*)

DELETE DRIVE 361  
 DELETE EVENT 362  
 DELETE EVENTSERVER 364  
 DELETE FILESPACE 365  
 DELETE GRPMEMBER 369  
 DELETE LIBRARY 371  
 DELETE MACHINE 372  
 DELETE MACHNODEASSOCIATION 373  
 DELETE MGMTCLASS 374  
 DELETE NODEGROUP 375  
 DELETE NODEGROUPMEMBER 376  
 DELETE PATH 377  
 DELETE POLICYSET 379  
 DELETE PROFASSOCIATION 380  
 DELETE PROFILE 383  
 DELETE RECMEDMACHASSOCIATION 385  
 DELETE RECOVERYMEDIA 386  
 DELETE SCHEDULE 387  
 DELETE SCRIPT 390  
 DELETE SERVER 391  
 DELETE SERVERGROUP 392  
 DELETE SPACETRIGGER 393  
 DELETE STGPOOL 394  
 DELETE SUBSCRIBER 395  
 DELETE SUBSCRIPTION 396  
 DELETE VIRTUALFSMAPPING 397  
 DELETE VOLHISTORY 398  
 DELETE VOLUME 403  
 deleted xxiv  
 DISABLE SESSIONS 411  
 DISMOUNT VOLUME 414  
 DISPLAY OBJNAME 415  
 DSMADMC 1  
 DSMSERV 1258  
 DSMSERV DISPLAY DBSPACE 1260  
 DSMSERV DISPLAY LOG 1261  
 DSMSERV FORMAT 1263  
 DSMSERV REMOVEDB 1269  
 DSMSERV RESTORE DB 1271, 1273  
 DSMSERV RUNFILE 1258  
 DSMULOG 1276  
 ENABLE EVENTS 417  
 ENABLE SESSIONS 420  
 END EVENTLOGGING 422  
 EXPIRE INVENTORY 424  
 EXPORT ADMIN 429  
 EXPORT NODE 435  
     directly to another server 445  
 EXPORT POLICY 453  
 EXPORT SERVER 459  
     directly to another server 467  
     to sequential media 461  
 EXTEND DB 474  
 GENERATE 475  
 GENERATE BACKUPSET 476  
 GENERATE BACKUPSETTOC 484  
 GRANT AUTHORITY 488  
 GRANT PROXYNODE 492  
 HALT 493  
 HELP 495  
 IDENTIFY DUPLICATES 497  
 IMPORT ADMIN 502  
 IMPORT NODE 505  
 IMPORT POLICY 511  
 IMPORT SERVER 514  
 INSERT MACHINE 519

commands (*continued*)

ISSUE MESSAGE 521  
 LABEL LIBVOLUME 523  
 LOCK ADMIN 530  
 LOCK NODE 531  
 LOCK PROFILE 532  
 MACRO 534  
 MIGRATE STGPOOL 536  
 MOVE DATA 540  
 MOVE DRMEDIA 544  
 MOVE GRPMEMBER 558  
 MOVE MEDIA 559  
 MOVE NODEDATA 567  
 new xxix  
 NOTIFY SUBSCRIBERS 576  
 PARALLEL 577  
 PING SERVER 579  
 PREPARE 580  
 QUERY ACTLOG 587  
 QUERY ADMIN 592  
 QUERY ASSOCIATION 596  
 QUERY AUDITOCUPANCY 598  
 QUERY BACKUPSET 601  
 QUERY BACKUPSETCONTENTS 606  
 QUERY CLOPTSET 609  
 QUERY COLLOCGROUP 611  
 QUERY CONTENT 614  
 QUERY COPYGROUP 621  
 QUERY DATAMOVER 626  
 QUERY DB 629  
 QUERY DBSPACE 632  
 QUERY DEVCLASS 633  
 QUERY DIRSPACE 637  
 QUERY DOMAIN 638  
 QUERY DRIVE 641  
 QUERY DRMEDIA 645  
 QUERY DRMSTATUS 654  
 QUERY ENABLED 657  
 QUERY EVENT 659  
 QUERY EVENTRULES 671  
 QUERY EVENTSERVER 673  
 QUERY EXPORT 674  
 QUERY FILESPACE 681  
 QUERY LIBRARY 686  
 QUERY LIBVOLUME 689  
 QUERY LICENSE 692  
 QUERY LOG 693  
 QUERY MACHINE 695  
 QUERY MEDIA 698  
 QUERY MGMTCLASS 704  
 QUERY MOUNT 707  
 QUERY NASBACKUP 708  
 QUERY NODE 712  
 QUERY NODEDATA 719  
 QUERY OCCUPANCY 724  
 QUERY OPTION 728  
 QUERY PATH 729  
 QUERY POLICYSET 733  
 QUERY PROCESS 736  
 QUERY PROFILE 738  
 QUERY PROXYNODE 741  
 QUERY RECOVERYMEDIA 742  
 QUERY REQUEST 745  
 QUERY RESTORE 746  
 QUERY RPFCONTENT 749  
 QUERY RPFFILE 751  
 QUERY SAN 754

commands (continued)

QUERY SCHEDULE 757  
 QUERY SCRIPT 765  
 QUERY SERVER 768  
 QUERY SERVERGROUP 772  
 QUERY SESSION 774  
 QUERY SHREDSTATUS 778  
 QUERY SPACETRIGGER 780  
 QUERY STATUS 782  
 QUERY STGPOOL 788  
 QUERY SUBSCRIBER 799  
 QUERY SUBSCRIPTION 801  
 QUERY SYSTEM 803  
 QUERY TAPEALERTMSG 805  
 QUERY TOC 806  
 QUERY VIRTUALFSMAPPING 809  
 QUERY VOLHISTORY 811  
 QUERY VOLUME 818  
 QUIT 825  
 RECLAIM STGPOOL 826  
 RECONCILE VOLUMES 829  
 REGISTER ADMIN 833  
 REGISTER LICENSE 835  
 REGISTER NODE 837  
 REMOVE ADMIN 848  
 REMOVE NODE 849  
 RENAME ADMIN 852  
 RENAME FILESPACE 853  
 RENAME NODE 856  
 RENAME SCRIPT 857  
 RENAME SERVERGROUP 858  
 RENAME STGPOOL 859  
 RESET PASSEXP 861  
 RESTORE NODE 866  
 RESTORE STGPOOL 871  
 RESTORE VOLUME 875  
 RESUME EXPORT 863  
 REVOKE AUTHORITY 881  
 REVOKE PROXYNODE 885  
 ROLLBACK 886  
 RUN 887  
 SELECT 890  
 SELECT, SQL xxxvii  
 SERIAL 897  
 SET ACCOUNTING 900  
 SET ACTLOGRETENTION 901  
 SET ARCHIVERETENTIONPROTECTION 903  
 SET AUTHENTICATION 905  
 SET CLIENTACTDURATION 906  
 SET CONFIGMANAGER 907  
 SET CONFIGREFRESH 908  
 SET CONTEXTMESSAGING 909  
 SET CROSSDEFINE 910  
 SET DBRECOVERY 911  
 SET DBREPORTMODE 912  
 SET DRMACTIVEDATASTGPOOL 913  
 SET DRMCHECKLABEL 915  
 SET DRMCMDFILENAME 916  
 SET DRMCOPYSTGPOOL 917  
 SET DRMCOURIERNAME 918  
 SET DRMDBBACKUPEXPIREDAYS 919  
 SET DRMFILEPROCESS 921  
 SET DRMINSTRPREFIX 922  
 SET DRMNOTMOUNTABLENAME 924  
 SET DRMPLANPREFIX 925  
 SET DRMPLANVPOSTFIX 927  
 SET DRMPRIMSTGPOOL 928

commands (continued)

SET DRMRPFEXPIREDAYS 929  
 SET DRMVaultNAME 931  
 SET EVENTRETENTION 932  
 SET INVALIDPWLIMIT 933  
 SET LICENSEAUDITPERIOD 934  
 SET MAXCMDRETRIES 935  
 SET MAXSCHEDSESSIONS 936  
 SET MINPWLENGTH 937  
 SET PASSEXP 938  
 SET QUERYSCHEDPERIOD 940  
 SET RANDOMIZE 941  
 SET REGISTRATION 943  
 SET RETRYPERIOD 945  
 SET SCHEDMODES 946  
 SET SERVERHLADDRESS 948  
 SET SERVERLLADDRESS 949  
 SET SERVERNAME 950  
 SET SERVERPASSWORD 951  
 SET SUMMARYRETENTION 953  
 SET TOCLOADRETENTION 955  
 SETOPT 956  
 SHRED DATA 958  
 SUSPEND EXPORT 960  
 UNLOCK ADMIN 962  
 UNLOCK NODE 963  
 UNLOCK PROFILE 964  
 UPDATE ADMIN 966  
 UPDATE BACKUPSET 968  
 UPDATE CLIENTOPT 973  
 UPDATE CLOPTSET 974  
 UPDATE COLLOCGROUP 975  
 UPDATE COPYGROUP 976  
 UPDATE DATAMOVER 984  
 UPDATE DEVCLASS 986  
 UPDATE DOMAIN 1053  
 UPDATE DRIVE 1055  
 UPDATE LIBRARY 1059  
 UPDATE LIBVOLUME 1064  
 UPDATE MACHINE 1066  
 UPDATE MGMTCLASS 1068  
 UPDATE NODE 1071  
 UPDATE NODEGROUP 1079  
 UPDATE PATH 1080  
 UPDATE POLICYSET 1085  
 UPDATE PROFILE 1087  
 UPDATE RECOVERYMEDIA 1088  
 UPDATE SCHEDULE 1090  
 UPDATE SCRIPT 1110  
 UPDATE SERVER 1112  
 UPDATE SERVERGROUP 1116  
 UPDATE SPACETRIGGER 1117  
 UPDATE STGPOOL 1119  
 UPDATE VIRTUALFSMAPPING 1153  
 UPDATE VOLHISTORY 1155  
 UPDATE VOLUME 1157  
 updated xxxi  
 VALIDATE LANFREE 1162  
 VALIDATE POLICYSET 1164  
 VARY 1166

Commands

DELETE KEYRING 370

commands in a macro

committing 84

rolling back 886

commands, administrative 8

commands, canceling 15

- COMMIT command 84
- committing commands in a macro 84
- COMMMETHOD option 1189
- COMMTIMEOUT option 1190
- communications, server-to-server 283
  - COMMMETHOD option 1189
  - shared memory between server and client 1189
- configuration manager 333
- configuration profile 254
- console mode
  - ending 2
  - restrictions 2
  - using 2
- continuation characters
  - for a list of values 11
  - for a quoted list of values 12
  - in output file 5
  - using the maximum number of 11
- conventions
  - typographic xvii
- COPIED parameter, QUERY CONTENT 616
- COPY ACTIVATEDATA command 86
- COPY CLOPTSET command 90
- COPY DOMAIN command 91
- copy group
  - defining archive 132
  - defining backup 127
  - deleting 356
  - description of 126
  - querying 621
  - restriction 126
  - updating archive 981
  - updating backup 977
- COPY MGMTCLASS command 93
- COPY POLICYSET command 95
- COPY PROFILE command 97
- COPY SCHEDULE command 99
- COPY SCRIPT 103
- COPY SERVERGROUP command 104
- COPYCONTINUE parameter
  - DEFINE STGPOOL, primary random access 299
  - DEFINE STGPOOL, primary sequential access 314
  - UPDATE STGPOOL, primary random access 1125
  - UPDATE STGPOOL, primary sequential access 1138
- copying
  - management class 93
  - policy domain 91
  - policy set 95
  - profile 97
  - schedule 99
  - script 103
  - server group 104
- COPYSTGPOLLS parameter
  - DEFINE STGPOOL, primary random access 298
  - DEFINE STGPOOL, primary sequential access 313
  - UPDATE STGPOOL, primary random access 1124
  - UPDATE STGPOOL, primary sequential access 1137
- COUNT parameter, QUERY CONTENT 615
- CRCDATA parameter
  - DEFINE STGPOOL, primary random access 296
  - DEFINE STGPOOL, primary sequential access 305, 323
  - UPDATE STGPOOL, primary random access 1122
  - UPDATE STGPOOL, primary sequential access 1131, 1145
- creating
  - backup set 476
  - client files onto a set of media 476

- customer support
  - contact xv

## D

- DAMAGED parameter, QUERY CONTENT 616
- data
  - moving 540
  - removing expired 424
- data deduplication 315, 324, 331, 1139, 1140, 1146, 1152
- data mover
  - defining 136
  - deleting 358
  - querying 626
  - updating 984
- data protection using WORM FILE volumes and SnapLock
  - when defining active-data pools 328
  - when defining copy storage pools 320
  - when defining sequential access storage pools 308
- data shredding, storage pools
  - backing up 56
  - defining 300
  - moving data 541
  - updating 1126
- database
  - backup 44
  - extending 474
  - installing 1263
  - querying 629
  - removing 1269
  - restoring 1271, 1273
  - setting options for 1172
  - transfer by data mover 136
- database recoverable space 629
- database recovery
  - back up volume history 58
  - delete volume history 398
  - query volume history 811
- database storage space 632, 1260
- DATAFORMAT parameter, define primary sequential access
  - storage pool 312, 323
- dataonly 6
- DATES parameter
  - IMPORT NODE 508
  - IMPORT SERVER 516
- DAYOFWEEK parameter
  - UPDATE SCHEDULE
    - client 1100
- DBMEMPERCENT option 1191
- debug ANR9999D message 909
- deduplicating data 497
- deduplication 315, 324, 331, 1139, 1140, 1146, 1152
- default management class, assigning 31
- DEFINE ASSOCIATION command 106
- DEFINE BACKUPSET command 108
- DEFINE CLIENTACTION command 113
- DEFINE CLIENTOPT command 118
- DEFINE CLOPTSET command 121
- DEFINE COLLOGGROUP command 122
- DEFINE COLLOCMEMBER command 124
- DEFINE COPYGROUP command 126
- DEFINE DATAMOVER command 136
- DEFINE DEVCLASS command 139
- DEFINE DOMAIN command 214
- DEFINE DRIVE command 216
- DEFINE EVENTSERVER command 220
- DEFINE GRPMEMBER command 221



- DEFINE LIBRARY command 222
- DEFINE MACHINE command 231
- DEFINE MACHNODEASSOCIATION command 233
- DEFINE MGMTCLASS command 235
- DEFINE PATH command 240
- DEFINE POLICYSET command 246
- DEFINE PROFASSOCIATION command 248
- DEFINE PROFILE command 254
- DEFINE RECMEDMACHASSOCIATION command 256
- DEFINE RECOVERYMEDIA command 258
- DEFINE SCHEDULE command 260
- DEFINE SCRIPT command 281
- DEFINE SERVER command 283
- DEFINE SERVERGROUP command 288
- DEFINE SPACETRIGGER command 289
- DEFINE STGPOOL command 292
- DEFINE SUBSCRIPTION command 333
- DEFINE VIRTUALFSMAPPING command 335
- DEFINE VOLUME command 337

#### defining

- association 106
- backup set 108
- client with schedule 261
- collocation group member 124
- collocation groups 122
- configuration manager 333
- copy group 126
- device class 139
- domain 214
- drive 216
- event server 220
- group member 221
- library 222
- management class 235
- node group 238
- node group member 239
- object with profile 248
- path for NAS data mover 240
- path for NDMP (NAS) connection 240
- policy set 246
- profile 254
- profile association 248
- schedule 260
- script 281
- server 283
- server group 288
- space trigger 289
- storage pool 292
- storage pool volume 337
- subscription 333

#### definitions 1285

- DELETE ASSOCIATION command 344
- DELETE BACKUPSET 346
- DELETE CLIENTOPT command 351
- DELETE CLOPTSET command 352
- DELETE COLLOGROUP command 353
- DELETE COLLOCMEMBER command 354
- DELETE COPYGROUP command 356
- DELETE DATAMOVER 358
- DELETE DEVCLASS command 359
- DELETE DOMAIN command 360
- DELETE DRIVE command 361
- DELETE EVENT command 362
- DELETE EVENTSERVER command 364
- DELETE FILESPACE command 365
- DELETE KEYRING command 370
- DELETE LIBRARY command 371

- DELETE MACHINE command 372
- DELETE MACHNODEASSOCIATION command 373
- DELETE MGMTCLASS command 374
- DELETE PATH 377
- DELETE POLICYSET command 379
- DELETE PROFASSOCIATION command 380
- DELETE PROFILE command 383
- DELETE RECMEDMACHASSOCIATION command 385, 916
- DELETE RECOVERYMEDIA command 386
- DELETE SCHEDULE command 387
- DELETE SCRIPT command 390
- DELETE SERVER command 391
- DELETE SERVERGROUP command 392
- DELETE SPACETRIGGER command 393
- DELETE STGPOOL command 394
- DELETE SUBSCRIBER command 395
- DELETE SUBSCRIPTION command 396
- DELETE VIRTUALFSMAPPING command 397
- DELETE VOLHISTORY command 398
- DELETE VOLUME command 403

#### deleting

- archive file deletion by a client, allowing 839, 1073
- backup file deletion by a client, allowing 840
- backup file deletion by a client, ing 1073
- backup set 346
- collocation group 353
- collocation group member 354
- copy group 356
- device class 359
- domain 360
- drive 361
- event record 362
- event server 364
- expired data 424
- file space 365
- group member 369
- library 371
- management class 374
- node group 375
- node group member 376
- policy set 379
- profile 383
- profile association 380
- schedule 387
- script 390
- server 391
- server group 392
- space trigger 393
- storage pool 394
- storage pool volume 403
- subscriber 395
- subscription 396
- volume history 398

#### determining retention periods for policy domains 638

#### DEVCONFIG option 1193

#### device class

- 3590 990
- 3592 147
- CENTERA 166, 1010
- defining 139
- deleting 359
- NAS 193, 1036
- OPTICAL 1038
- QIC 200, 1042
- querying 633
- updating 986
- VOLSAFE 74, 210, 1050

- device class *(continued)*
  - WORM 1038
- device configuration file 47
- device drivers xx
- device names
  - for devices connected to NAS file server 240, 1081
- device type
  - 3570 140, 987
  - 3590 143, 990
  - 3592 147, 994
  - 4MM 152, 998
  - 8MM 156, 1002
  - CARTRIDGE 162, 1007
  - CENTERA 166, 1010
  - DLT 168, 1012, 1017
  - DTF 174
  - ECARTRIDGE 177, 1020
  - FILE 182, 637, 1025
  - GENERICTAPE 185, 1029
  - LTO 188, 1031
  - NAS 193, 1036
  - OPTICAL 196, 1038
  - QIC 200, 1042
  - REMOVABLEFILE 205, 1046
  - SERVER 208, 1048
  - VOLSAFE 210, 1050
  - WORM 196, 1038
  - WORM12 196, 1038
  - WORM14 196, 1038
- directories, specifying multiple for FILE volumes 183, 1026
- DISABLE EVENTS 407
- DISABLE EVENTS command 407
- DISABLE SESSIONS command 411
- DISABLESCHEDS option 1194
- disabling
  - context messaging for ANR9999D messages 909
  - server 411
- DISCARDATA parameter 404
- disk space, migrating data to create 297, 537, 1122
- disk volume performance 292
- disk volumes 292
- disk-only backup
  - defining FILE device classes 183
  - defining volumes 337
  - updating FILE device classes 1026
- DISKMAP option 1196
- DISKSTGPOOLMEMSIZE option 1194
- DISMOUNT VOLUME command 414
- dismounting volumes 414
- DISPLAY OBJNAME command 415
- displaying
  - full object name 415
  - output 6
  - product version information 6
  - scheduled and completed events 659
- DISPLAYLFINFO option 1195
- DLT
  - device type 168
  - WORM media 169
- domain
  - associating with profile 248
  - copy 91
  - defining 214
  - deleting 360
  - distributing 254
  - querying 638
  - updating 1053
- drive
  - cleaning 83, 218
  - defining 216, 641
  - deleting 361
  - querying 641
  - updating 1055
- drive encryption
  - 3592 device class 151, 997
  - LTO device class 192, 1034
- DRIVEACQUIRERETRY option 1197
- DRIVEENCRYPTION parameter
  - 3592 device class 151, 997
  - LTO device class 192, 1034
- DSMADMC command 1
- DSMADMC options 5
- DSMSERV command 1258
- DSMSERV DISPLAY DBSPACE command 1260
- DSMSERV DISPLAY LOG command 1261
- DSMSERV FORMAT command 1263
- DSMSERV INSERTDB
  - preview xx
- DSMSERV REMOVEDB command 1269
- DSMSERV RESTORE DB command 1271
- DSMSERV RUNFILE command 1258
- DSMULOG command 1276
- duplicate-identification processes 315, 324, 331, 1140, 1146, 1152
- DURATION parameter
  - DEFINE SCHEDULE, administrative command 276
  - DEFINE SCHEDULE, client 268
  - SHRED DATA, administrative command 958
  - UPDATE SCHEDULE, administrative command 1106
  - UPDATE SCHEDULE, client 1097

## E

- ECARTRIDGE device type 177
- education
  - see Tivoli technical training xiv
- ENABLE EVENTS command 417
- ENABLE SESSIONS command 420
- enabling
  - context messaging for ANR9999D 909
  - server 420
- encryption
  - application
    - 3592 151, 997
    - LTO 192, 1034
  - drive
    - 3592 151, 997
    - LTO 192, 1034
  - library 151, 997
  - system 151, 997
- END EVENTLOGGING command 422
- ENDDATE parameter 588, 662, 709
- ending
  - batch mode 3
  - console mode 2
  - interactive mode 2
  - mount mode 2
- ENDTIME parameter 589, 662, 710
- error
  - ANR9999D message 909
  - file specification 116, 266, 1095
  - In Progress status for QUERY EVENT 665



- ESTCAPACITY parameter
  - DEFINE DEVCLASS
    - 3570 141
    - 3590 145
    - 3592 149
    - 4MM 154
    - 8MM 159
    - CARTRIDGE 163
    - DLT 171
    - DTF 175
    - ECARTRIDGE 180
    - GENERICTAPE 186
    - LTO 190
    - NAS 194
    - Optical/WORM 198
    - QIC 202
    - VOLSAFE 211
  - UPDATE DEVCLASS
    - 3570 988
    - 3590 991
    - 3592 995
    - 4MM 999
    - 8MM 1005
    - CARTRIDGE 1008
    - DLT 1015
    - DTF 1018
    - ECARTRIDGE 1022
    - GENERICTAPE 1029
    - NAS 1037
    - Optical/WORM 1040
    - QIC 1044
    - VOLSAFE 1051
- event logging
  - send events to receivers 60
  - set logging off by receivers 422
- event record
  - description of 362, 932
  - managing 932
  - querying 659
  - removing from the database 362
  - setting retention period 932
- event record retention period
  - managing 932
  - querying 782
- event, querying 659
- EVENTSERVER option 1198
- EXCEPTIONSONLY parameter 662
- EXPINTERVAL option 1199
- EXPIRATION parameter
  - DEFINE SCHEDULE, administrative command 272
  - DEFINE SCHEDULE, client 261
  - UPDATE SCHEDULE, administrative command 1102
  - UPDATE SCHEDULE, client 1091
- EXPIRE INVENTORY command 424
- expiring inventory 424
- export
  - administrator 429
  - node 435
  - policy information 453
  - server 459
- EXPORT ADMIN
  - information directly to another server 433
  - to sequential media 430
- EXPORT ADMIN command 429
- EXPORT NODE
  - export node definitions to sequential media 437
- EXPORT NODE command 435

- EXPORT POLICY command 453
- EXPORT SERVER command 459
- EXPQUIET option 1200
- EXTEND DB command 474
- extending
  - database 474

## F

- failover directory for the archive log 1261
- FILE device type
  - concurrent read/write access to FILE volumes
    - increasing number of client mount points 841, 1074
    - visible as output in QUERY MOUNT 707
    - visible as output in QUERY SESSION 774
  - creating and formatting private volumes 337
  - defining the device class 182
  - updating the device class 1025
- file server, NAS
  - names for connected devices 240, 1081
  - node 136, 840
- file space
  - automatic rename for Unicode support 1074
  - deleting 365
  - FSID 854
  - hexadecimal representation 854
  - querying 681
  - renaming 853
  - specification errors, restrictions 114, 264, 1094
- FILEDATA parameter
  - EXPORT NODE 439, 447
  - IMPORT NODE 507
- FILEEXIT option 1201
- FILENAMES parameter
  - BACKUP DEVCONFIG 47
  - BACKUP VOLHISTORY 59
- files
  - collocating by client 309, 319, 326
  - moving 540
  - removing expired 424
- FILESPEC parameter
  - EXPORT NODE 438, 446
  - IMPORT NODE 506
- FILETEXTEXIT option 1202
- firewall, opening server ports
  - administrative sessions 1240
  - client sessions 1243
  - TCP/IP ports 1240, 1243
- fixes, obtaining xv
- FORCE parameter, CHECKOUT LIBVOLUME 81
- FORMAT parameter
  - DEFINE DEVCLASS
    - 3570 140
    - 3590 143
    - 3592 148
    - 4MM 153
    - 8MM 157
    - CARTRIDGE 162
    - DLT 169
    - DTF 174
    - ECARTRIDGE 178
    - LTO 189
    - Optical/WORM 197
    - QIC 201
    - VOLSAFE 211
  - UPDATE DEVCLASS
    - 3570 987

FORMAT parameter *(continued)*  
UPDATE DEVCLASS *(continued)*

3590 990  
3592 994  
4MM 998  
8MM 1002  
CARTRIDGE 1007  
DLT 1013  
DTF 1017  
ECARTRIDGE 1021  
LTO 1032  
QIC 1039, 1043  
VOLSAFE 1050

FORMAT, DSMSEV 1263

FREQUENCY parameter

DEFINE COPYGROUP, archive 133  
DEFINE COPYGROUP, backup 128  
UPDATE COPYGROUP, archive 981  
UPDATE COPYGROUP, backup 978

full object name

displaying 415

## G

GENERATE BACKUPSET command 476

GENERATE BACKUPSETTOC 484

GENERICTAPE device type 185, 1029

globally unique identifier (GUID) 717

glossary 1285

GRANT AUTHORITY command 488

GRANT PROXYNODE command 492

granting

authority 488

types of privilege classes 488

group, collocation by

defining a group 122

defining a group member 124

deleting a group 353

deleting a group member 354

querying a group 611

updating a group 975

group, node

defining 238

defining a member 239

deleting a group 375

deleting a member 376

querying a group 722

updating a group 1079

GUID 717

## H

HALT command 493

halting the server 493

HELP command 495

help for administrative commands 495

hierarchical storage management

DEFINE MGMTCLASS 235

description of 235

UPDATE MGMTCLASS 1068

hierarchy of storage pools

considerations for backup 54

defining 292

high level address, Centera 166, 1010

HIGHMIG parameter

DEFINE STGPOOL

primary random access 297

primary sequential access 306

UPDATE STGPOOL

primary random access 1122

primary sequential access 1132

history information

backing up 58

deleting 398

querying 811

HSM (hierarchical storage management) 236, 1068

## I

IBM Software Support

submitting a problem xvii

IBM Support Assistant xiv

IDENTIFY DUPLICATES command 497

IDLETIMEOUT option 1203

IMPORT ADMIN command 502

IMPORT NODE command 505

IMPORT POLICY command 511

IMPORT SERVER command 514

importing

administrator 502

node 505

policy information 511

server 514

In Progress status for QUERY EVENT command 665

incremental backup of database 44

inheritance, copy storage pool lists and

defining random access storage pools 299

defining sequential access storage pools 313

updating random access storage pools 1125

updating sequential access storage pools 1138

INSERT MACHINE command 519

installing

database 1263

recovery log 1263

interactive mode

continuation characters 11

ending 2

quitting 825

restrictions 3

using 3

Internet, searching for problem resolution xiv, xv

inventory expiring 424

IP address, Centera 166, 1010

ISSUE MESSAGE command 521

issuing commands for a one-time action 113

ITEMCOMMIT option 6

## K

knowledge bases, searching xiv

## L

label

library volume 523

LAN-free data movement

validating 1162

language limitations 1

LANGUAGE option 1204

- library
  - auditing 34
  - defining 222
  - deleting 371
  - querying 686
  - shared 227, 1062
  - updating 1059
- library volume
  - checking in 70
  - checking out 77
  - labeling 523
  - querying 689
  - updating 1064
- license
  - auditing 36
  - querying 692
  - registering 835
  - setting audit period 934
- licensing xix
- LOCK ADMIN command 530
- LOCK NODE command 531
- LOCK PROFILE command 532
- locking
  - administrator 530
  - node 531
  - profile 532
- logic flow statements in scripts 1277
- logs, recovery 1261
- LOWMIG parameter
  - DEFINE STGPOOL
    - primary random access 297
    - primary sequential access 306
  - UPDATE STGPOOL
    - primary random access 1123
    - primary sequential access 1132
- LTO
  - device class
    - defining 188
    - updating 1031
  - WORM 189

## M

- MACRO command 534
- macros
  - continuation characters 11
  - rolling back 886
  - using 534
- managed server 333
- managed system for storage area network (SAN)
  - DEFINE LIBRARY command 222
  - UPDATE LIBRARY command 1059
- management class
  - copying 93
  - defining the default 31
  - deleting 374
  - querying 704
  - updating 1068
- MAXCAPACITY parameter
  - DEFINE DEVCLASS
    - FILE 183
    - REMOVABLEFILE 206
    - SERVER 208
  - UPDATE DEVCLASS
    - FILE 1025
    - REMOVABLEFILE 1046
    - SERVER 1048
- maximum retries, setting 935
- MAXPROCESS parameter
  - BACKUP STGPOOL 55
  - UPDATE STGPOOL 873
- MAXSCRATCH parameter
  - DEFINE STGPOOL
    - active-data pool 330
    - copy sequential access 322
    - primary sequential access 310
  - UPDATE STGPOOL
    - active-data pool 1150
    - copy sequential access 1144
    - primary sequential access 1135
- MAXSESSIONS option 1206
- MAXSIZE parameter
  - DEFINE STGPOOL
    - primary random access 296
    - primary sequential access 305
  - UPDATE STGPOOL
    - primary random access 1121
    - primary sequential access 1130
- media mount requests 67
- media support, CD 205
- member, server group 221
- MERGEFILESPPACES parameter
  - EXPORT NODE 451
  - EXPORT SERVER 472
  - IMPORT NODE 510
  - IMPORT SERVER 514
- message diagnosis, ANR9999D 909
- MESSAGEFORMAT option 1207
- MIGDESTINATION parameter
  - DEFINE MGMTCLASS 236
  - UPDATE MGMTCLASS 1069
- MIGPROCESS parameter
  - DEFINE STGPOOL, primary random access 297
  - UPDATE STGPOOL, primary random access 1123
- MIGRATE STGPOOL command 536
- migration
  - files from client node 236
  - specifying multiple concurrent processes
    - using the DEFINE STGPOOL command 311
    - using the UPDATE STGPOOL command 1136
  - starting manually 536
  - storage pool, random access
    - high migration threshold 297, 1122
    - low migration threshold 297, 537, 1123
  - storage pool, sequential access
    - high migration threshold 306, 1132
    - low migration threshold 306, 537, 1132
- MIGREQUIRESBKUP parameter
  - DEFINE MGMTCLASS 236
  - UPDATE MGMTCLASS 1069
- MIRRORLOGDIR option 1208
- MODE parameter
  - DEFINE COPYGROUP, archive 134
  - DEFINE COPYGROUP, backup 129
  - UPDATE COPYGROUP, archive 982
  - UPDATE COPYGROUP, backup 979
- monitoring
  - Tivoli Storage Manager activities 2
- monitoring Tivoli Storage Manager activities 2
- mount mode
  - ending 2
  - using 2
- mount request 67
  - canceled 67

mount request *(continued)*  
     querying 745  
 mounted sequential access volumes  
     dismounting 414  
     querying 707  
 MOUNTLIMIT parameter  
     DEFINE DEVCLASS  
         3570 142  
         3590 145  
         3592 150  
         4MM 155  
         8MM 156  
         CARTRIDGE 164  
         CENTERA 167  
         DLT 172  
         DTF 176  
         ECARTRIDGE 181  
         GENERICTAPE 186  
         LTO 191  
         NAS 194  
         Optical/WORM 199  
         QIC 203  
         REMOVABLEFILE 206  
         SERVER 209  
         VOLSAFE 212  
     UPDATE DEVCLASS  
         3570 989  
         3590 990  
         3592 996  
         4MM 1000  
         8MM 1002  
         CARTRIDGE 1009  
         CENTERA 1011  
         DLT 1015  
         DTF 1019  
         ECARTRIDGE 1023  
         FILE 1025  
         GENERICTAPE 1030  
         LTO 1034  
         Optical/WORM 1041  
         QIC 1045  
         REMOVABLEFILE 1047  
         VOLSAFE 1052  
 MOUNTRETENTION parameter  
     DEFINE DEVCLASS  
         3570 142  
         3590 145  
         3592 150  
         4MM 154  
         8MM 160  
         CARTRIDGE 164  
         DLT 172  
         DTF 176  
         ECARTRIDGE 181  
         GENERICTAPE 186  
         LTO 191  
         NAS 193  
         Optical/WORM 199  
         QIC 203  
         REMOVABLEFILE 206  
         SERVER 209  
         VOLSAFE 212  
     UPDATE DEVCLASS  
         3570 988  
         3590 992  
         3592 996  
         4MM 1000  
         8MM 1005  
         CARTRIDGE 1009  
         DLT 1015  
         DTF 1018  
         ECARTRIDGE 1023  
         GENERICTAPE 1030  
         LTO 1034  
         NAS 1036  
         Optical/WORM 1040  
         QIC 1045  
         REMOVABLEFILE 1046  
         SERVER 1049  
         VOLSAFE 1052  
 MOVE DATA command 540  
 MOVE DRMEDIA command 544  
 MOVE GRPMEMBER command 558  
 MOVE MEDIA command 559  
 MOVE NODEDATA command 567  
 MOVEBATCHSIZE option 1209  
 MOVESIZETHRESH option 1210  
 moving  
     data in storage pool 540  
     files 540  
     group member 558  
 MSGINTERVAL option 1211  
 MSGNO parameter, QUERY ACTLOG 589  
 MOUNTRETENTION parameter *(continued)*  
 UPDATE DEVCLASS *(continued)*  
     8MM 1005  
     CARTRIDGE 1008  
     DLT 1015  
     DTF 1018  
     ECARTRIDGE 1023  
     GENERICTAPE 1029  
     LTO 1033  
     NAS 1036  
     Optical/WORM 1040  
     QIC 1045  
     REMOVABLEFILE 1046  
     SERVER 1049  
     VOLSAFE 1051  
 MOUNTWAIT parameter  
     DEFINE DEVCLASS  
         3570 142  
         3590 145  
         3592 150  
         4MM 154  
         8MM 160  
         CARTRIDGE 164  
         DLT 172  
         DTF 176  
         ECARTRIDGE 181  
         GENERICTAPE 186  
         LTO 191  
         NAS 194  
         Optical/WORM 199  
         QIC 203  
         REMOVABLEFILE 206  
         VOLSAFE 212  
     UPDATE DEVCLASS  
         3570 988  
         3590 992  
         3592 996  
         4MM 1000  
         8MM 1005  
         CARTRIDGE 1009  
         DLT 1015  
         DTF 1018  
         ECARTRIDGE 1023  
         GENERICTAPE 1030  
         LTO 1034  
         NAS 1036  
         Optical/WORM 1040  
         QIC 1045  
         REMOVABLEFILE 1046  
         SERVER 1049  
         VOLSAFE 1052

## N

- name
  - devices 240
  - for device connected to NAS file server 240, 1081
- naming conventions
  - introduction 12
  - naming volumes 12
  - passwords 12
  - restrictions 12
- NAS file server
  - data mover, defining 136
  - name for connected device 240, 1081
  - node name 136
  - path, defining 240
- NDMPCONTROLPORT option 1212
- NDMPPORTRANGE option 1213
- NDMPPREFDATAINTERFACE option 1214
- NetApp DataONTAP, for managing FILE volumes
  - when defining active-data pools 328
  - when defining copy storage pools 321
  - when defining sequential access storage pools 308
- network-attached storage (NAS) file server
  - data mover, defining 136
  - device class 193, 1036
  - path, defining 240
- NEXTSTGPOOL parameter
  - DEFINE STGPOOL
    - primary random access 296
    - primary sequential access 305
  - UPDATE STGPOOL
    - primary random access 1122
    - primary sequential access 305, 1131
- node
  - associating with a schedule 106
  - exporting 435
  - finding file for 724
  - importing 505
  - locking 531
  - name for data mover 136
  - NAS file server 136, 840
  - node group
    - defining a group 238
    - defining a member 239
    - deleting a group 375
    - deleting a member 376
    - querying a group 722
    - updating a group 1079
  - querying 712
  - registering 837
  - removing 849
  - renaming 856
  - unlocking 963
  - updating 1071
- NOPREEMPT option 1215
- NORETRIEVEDATE option 1216
- normal recovery log mode 44
- NOTIFY SUBSCRIBERS command
  - example 576
  - related commands 576
- NUMOPENVOLSALLOWED option 1217

## O

- OBJECTS parameter
  - DEFINE SCHEDULE, client 268
  - UPDATE SCHEDULE, client 1097

- occupancy
  - displaying file space information 724
  - querying 724
- offsite volumes, specifying the number to be reclaimed
  - using the DEFINE STGPOOL command 321, 329
  - using the UPDATE STGPOOL command 1144, 1150
- OPTICAL device type 196, 1038
- option set 118
- options
  - changes in V6 xxix
  - deleted xxxiv
  - new xxix
  - updated xxxi
- OPTIONS parameter
  - DEFINE SCHEDULE, client 268
  - UPDATE SCHEDULE, client 1097
- options, client 118
- options, server
  - importance for recovery 44
  - querying 728
- output
  - displaying 6
- output headers
  - displaying 6
- overflow location for logs 1261
- OWNER parameter, DELETE FILESPACE 367

## P

- PARALLEL command 577
- parameters, descriptions in 14
- parameters, entering 10
- password
  - case-sensitivity 12
  - characters allowed for entering 12
  - entered with the REGISTER ADMIN command 833
  - entered with the REGISTER NODE command 837
  - entered with the UPDATE ADMIN command 966
  - setting expiration interval 938
  - setting the maximum length 12
- path
  - defining 240
  - deleting 377
  - querying 729
  - updating 1080
- pattern-matching expression, creating 13
- performance
  - improving communications with client using shared memory 1189, 1231
  - limiting offsite volumes to be reclaimed
    - using the DEFINE STGPOOL command 321, 329
    - using the UPDATE STGPOOL command 1144, 1150
  - specifying multiple concurrent migration processes
    - using the DEFINE STGPOOL command 311
    - using the UPDATE STGPOOL command 1136
  - specifying multiple concurrent reclamation processes
    - using the DEFINE STGPOOL command 320, 328
    - using the UPDATE STGPOOL command 1143, 1149
  - storage requirements, reducing by deduplication 315, 324, 331, 1139, 1140, 1146, 1152
- PERUNITS parameter
  - DEFINE SCHEDULE, administrative command 278
  - UPDATE SCHEDULE, administrative command 1108
  - UPDATE SCHEDULE, client 1100
- policy domain
  - associating with profile 248
  - copy 91

- policy domain (*continued*)
  - defining 214
  - deleting 360
  - distributing 254
  - querying 638
  - updating 1053
- policy privilege class
  - privilege class
    - policy 21
  - restricted 21
  - unrestricted 21
- policy set
  - activating 29
  - copying 95
  - defining 246
  - deleting 379
  - querying 733
  - updating 1085
  - validating 1164
- polling
  - information about 940
  - setting frequency 940
- Pool Entry Authorization (PEA) file, Centera 166, 1010
- ports for firewall
  - administrative sessions 1240
  - client sessions 1243
  - TCP/IP ports 1240, 1243
- preempting of operations 1215
- PREFIX parameter
  - DEFINE DEVCLASS
    - 3570 141
    - 3590 145
    - 3592 150
    - 4MM 154
    - 8MM 160
    - CARTRIDGE 163
    - DLT 172
    - DTF 175
    - ECARTRIDGE 180
    - LTO 191
    - NAS 194
    - QIC 203
    - SERVER 209
    - VOLSAFE 212
  - UPDATE DEVCLASS
    - 3570 988
    - 3590 992
    - 3592 996
    - 4MM 1000
    - 8MM 1005
    - CARTRIDGE 1008
    - DLT 1015
    - DTF 1018
    - ECARTRIDGE 1023
    - LTO 1033
    - NAS 1036, 1037
    - QIC 1044
    - SERVER 1048
    - VOLSAFE 1051
- PREPARE command 580
- PREVIEW parameter
  - EXPORT NODE 439, 508
- PREVIEWIMPORT parameter
  - EXPORT ADMIN 433
  - EXPORT NODE 451
  - EXPORT POLICY 457
  - EXPORT SERVER 472
- printing redirected output 4
- privilege class
  - administrator issued 25
  - none required 25
  - operator 23
  - policy 22
  - restricted 22
  - storage 22, 23
  - system 18
  - unrestricted 22
- problem determination
  - describing problem for IBM Software Support xvi
  - determining business impact for IBM Software Support xvi
  - submitting a problem to IBM Software xvii
- process
  - canceling 65
  - querying 736
- product version information
  - displaying 6
- profile 254
- profile association 248
- protecting WORM volumes with SnapLock
  - when defining active-data pools 328
  - when defining copy storage pools 320
  - when defining sequential access storage pools 308
- publications
  - download xi
  - order xi
  - related hardware xiii
  - search xi
  - Tivoli Storage Manager xi

## Q

- QIC device type 200, 1042
- QUERY ACTLOG command 587
- QUERY ADMIN command 592
- QUERY ASSOCIATION command 596
- QUERY AUDITOCUPANCY command 598
- QUERY BACKUPSET 601
- QUERY BACKUPSETCONTENTS 606
- QUERY CLOPTSET command 609
- QUERY COLLOGGROUP command 611
- QUERY CONTENT command 614
- QUERY COPYGROUP command 621
- QUERY DATAMOVER command 626
- QUERY DB command 629
- QUERY DBSPACE command 632
- QUERY DEVCLASS command 633
- QUERY DOMAIN command 638
- QUERY DRIVE command 641
- QUERY DRMEDIA command 645
- QUERY DRMSTATUS command 654
- QUERY ENABLED command 657
- QUERY EVENT command 659
- QUERY EVENTRULES command 671, 673
- QUERY FILESPACE command 681
- QUERY LIBRARY command 686
- QUERY LIBVOLUME command 689
- QUERY LICENSE command 692
- QUERY LOG command 693
- QUERY MACHINE command 695
- QUERY MEDIA command 698
- QUERY MGMTCLASS command 704
- QUERY MOUNT command 707
- QUERY NASBACKUP command 708



- QUERY NODE command 712
- QUERY NODEDATA command 719
- QUERY OCCUPANCY command 724
- QUERY OPTION command 728
- QUERY PATH command 729
- QUERY POLICYSET command 733
- QUERY PROCESS command 736
- QUERY PROFILE command 738
- QUERY PROXYNODE command 741
- QUERY RECOVERYMEDIA command 742
- QUERY REQUEST command 745
- QUERY RESTORE command 746
- QUERY RPFCONTENT command 749
- QUERY RPFILE command 751
- QUERY SAN command 754
- QUERY SCHEDULE command 757
- QUERY SCRIPT command 765
- QUERY SERVER command 768
- QUERY SERVERGROUP command 772
- QUERY SESSION command 774
- QUERY SHREDSTATUS command 778
- QUERY SPACETRIGGER command 780
- QUERY STATUS command 782
- QUERY STGPOOL command 788
- QUERY SUBSCRIBER command 799
- QUERY SUBSCRIPTION command 801
- QUERY SYSTEM command 803
- QUERY TAPEALERTMSG command 805
- QUERY TOC command 806
- QUERY VIRTUALFSMAPPING command 809
- QUERY VOLHISTORY command 811
- QUERY VOLUME command 818
- QUERYAUTH server option 1219
- querying
  - activity log 587
  - administrator 592
  - audit occupancy 598
  - backup sets 601
  - clients with schedules 596
  - completed events 659
  - contents of a backup set 606
  - contents of a volume 614
  - copy group 621
  - database 629
  - device class 633
  - domain 638
  - drive 641
  - event server 673
  - file space 681
  - library 686
  - library volume 689
  - license 692
  - management class 704
  - mount requests 745
  - node 712
  - node groups 722
  - occupancy 724
  - option 728
  - policy set 733
  - process 736
  - profile 738
  - recovery log 693
  - recovery plan file 751
  - recovery plan file content 749
  - scheduled events 659
  - schedules 757
  - script 765

- querying (*continued*)
  - server group 772
  - session 774
  - space trigger 780
  - status 782
  - storage pool 788
  - storage pool volume 818
  - subscriber 799
  - subscription 801
  - volume history file 811
- QUIT command 825
- quitting
  - batch mode 3
  - interactive mode 2, 825
  - mount mode 2

## R

- random access volumes 292
- raw partitions, performance effect 292
- RECLAIM parameter
  - DEFINE STGPOOL
    - active-data pool 327
    - copy sequential access 319
    - primary sequential access 306
  - UPDATE STGPOOL
    - active-data pool 1149
    - copy sequential access 1143
    - primary sequential access 1132
- RECLAIM STGPOOL command 826
- RECLAIMDELAY option 1220
- RECLAIMPERIOD option 1221
- reclamation of volumes
  - specifying a threshold of reclaimable space
    - using the DEFINE STGPOOL command 307, 319, 327
    - using the UPDATE STGPOOL command 1132, 1143, 1149
  - specifying multiple concurrent processes
    - using the DEFINE STGPOOL command 320, 328
    - using the UPDATE STGPOOL command 1143, 1149
  - starting manually 826
- RECLAMATIONTYPE parameter
  - defining active-data pools 328
  - defining copy storage pools 320
  - defining sequential access storage pools 308
- RECONCILE VOLUMES command 829
- recovery log
  - installing 1263
  - setting options for 1172
- recovery logs 1261
- recovery plan file
  - display contents 749
  - query information 751
  - set expire days 929
- recovery plan prefix 925
- redirecting command output 4
- redirection characters
  - types of 4
  - using 4
- REGISTER ADMIN command 833
- REGISTER LICENSE command 835
- REGISTER NODE command 837
- registering
  - administrator 833
  - license 835
  - node 837

- registration
  - closed 943
  - description of 943
  - setting 943
- REMOVABLEFILE device type 205, 1046
- REMOVABLEFILE, CD support 205
- REMOVE ADMIN command 848
- REMOVE NODE command 849
- removing
  - administrator 848
  - client association 344
  - node 849
- RENAME ADMIN command 852
- RENAME FILESPACE command 853
- RENAME NODE command 856
- RENAME SCRIPT command 857
- RENAME SERVERGROUP command 858
- RENAME STGPOOL command 859
- renaming
  - administrator 852
  - file space 853
  - node 856
  - script 857
  - server group 858
  - storage pool 859
- REPLACEDFS parameter
  - EXPORT ADMIN 433
  - EXPORT NODE 451
  - EXPORT POLICY 457
  - EXPORT SERVER 472
  - IMPORT ADMIN 503
  - IMPORT NODE 509
  - IMPORT POLICY 512
- REPLY command 860
- reporting error message for ANR9999D 909
- REPORTRETRIEVE server option 1222
- REQSYSAUTHOUTFILE option 1223
- request, mount 67, 745
- RESET PASSEXP command 861
- RESOURCETIMEOUT option 1224
- RESTORE DB, DSMSEV 1270
- RESTORE NODE command 866
- RESTORE STGPOOL command 871
- RESTORE VOLUME command 875
- RESTOREINTERVAL option 1225
- restoring
  - as action on client command 114, 264, 1094
- retention period
  - description of 932
  - setting 932
- retention period, event record
  - managing 932
  - querying 782
- RETENTIONEXTENSION option 1226
- RETONLY parameter
  - DEFINE COPYGROUP, backup 129
  - UPDATE COPYGROUP, backup 979
- retry period
  - description of 945
  - setting 945
- return code checking 1277
- RETVAR parameter
  - DEFINE COPYGROUP, archive 133
  - UPDATE COPYGROUP, archive 981
- REUSEDELAY parameter
  - DEFINE STGPOOL
    - active-data pool 330

- REUSEDELAY parameter (*continued*)
    - DEFINE STGPOOL (*continued*)
      - copy sequential access 322
      - primary sequential access 310
    - UPDATE STGPOOL
      - active-data pool 1151
      - copy sequential access 1145
      - primary sequential access 1135
  - reusing volumes
    - active-data pool 327, 1149
    - copy sequential access 319, 1143
    - primary sequential access 307, 1132
  - REVOKE AUTHORITY command 881
  - REVOKE PROXYNODE command 885
  - revoking
    - authority 881
    - types of privilege classes 881
  - roll-forward recovery log mode 44
  - ROLLBACK command 886
  - rolling back commands in a macro 886
  - routing commands 16
  - RUN command 887

## S

- SAN
  - DEFINE LIBRARY command 222
  - UPDATE LIBRARY command 1059
- SAN tape devices 226, 228
- SANDISCOVERYTIMEOUT option 1228
- SANDISCOVERYTIMEOUT server option xx
- SANREFRESHTIME server option 1229
- schedule
  - administrative command 272
  - associating with a client node 106
  - client 261, 272
  - copying 99
  - defining 235, 260
  - deleting 387
  - description of 272
  - querying 757
  - querying results of (events) 659
  - restrictions of 261
  - startup window, defining schedule 268, 276
  - startup window, updating schedule 1097, 1106
  - types of 260
  - updating 1090
- schedule event
  - querying 659
  - setting start date for displaying 660
  - setting start time for displaying 661, 662, 667, 668
  - viewing information about 659
- scheduling mode
  - information about 946
  - setting 946
- scratch volumes in storage pool
  - defining a storage pool 310, 322, 330
  - updating a storage pool 1135, 1144, 1150
- scripts
  - copying 103
  - defining 281
  - deleting 390
  - querying 765
  - renaming 857
  - running 887
  - running commands in parallel 577
  - running commands serially 897



- scripts (*continued*)
  - updating 1110
- SEARCHMPQUEUE option 1230
- security, encryption
  - 3592 devices 151, 997
  - LTO devices 192, 1034
- SELECT command xxxvii, 890
- sequential volume history
  - backing up 58
  - deleting 398
  - display sequential volume history 811
  - querying 811
- SERIAL command 897
- SERIALIZATION parameter
  - DEFINE COPYGROUP, archive 134
  - DEFINE COPYGROUP, backup 129
  - UPDATE COPYGROUP, archive 982
  - UPDATE COPYGROUP, backup 979
- server
  - commands
    - changes in V6 xxix
    - deleted xxxiv
    - new xxix
    - updated xxxi
  - deleted commands xxxiv
  - deleted options xxxiv
  - deleted utilities xxxiv
  - disabling 411
  - enabling 420
  - exporting 459
  - importing 514
  - migrating 536
  - new commands xxix
  - new options xxix
  - new utilities xxix
  - options
    - changes in V6 xxix
    - deleted xxxiv
    - new xxix
    - updated xxxi
  - reclaiming 826
  - setting name for 950
  - updated commands xxxi
  - updated options xxxi
  - updated utilities xxxi
  - utilities
    - changes in V6 xxix
    - deleted xxxiv
    - new xxix
    - updated xxxi
- server console
  - restrictions 7
  - using 7
- server name settings 950
- server options
  - 3494SHARED 1175
  - ACSACCESSID 1176
  - ACSLOCKDRIVE 1177
  - ACSQUICKINIT 1178
  - ACSTIMEOUTX 1179
  - ACTIVELOGDIR 1180
  - ACTIVELOGSIZE 1181
  - ADMINONCLIENTPORT 1182
  - AIXASYNCIO 1183
  - ALIASHALT 1184
  - ARCHFAILOVERLOGDIR 1185
  - ARCHIVELOGDIR 1186

- server options (*continued*)
  - ASSISTVCRRECOVERY 1187
  - AUDITSTORAGE 1188
  - changing with SETOPT command 728
  - COMMMETHOD 1189
  - COMMTIMEOUT 1190
  - DBMEMPERCENT 1191
  - deleted xxxiv
  - DEVCONFIG 1193
  - DISABLESCHEDS 1194
  - DISKMAP 1196
  - DISKSTGPOOLMEMSIZE 1194
  - DISPLAYLFINFO 1195
  - DRIVEACQUIRERETRY 1197
  - EVENTSERVER 1198
  - EXPINTERVAL 1199
  - EXPQUIET 1200
  - FILEEXIT 1201
  - FILETEXTEXIT 1202
  - IDENTIFYAUTOSTART 1192
  - IDLETIMEOUT 1203
  - LANGUAGE 1204
  - MAXSESSIONS 1206
  - message options 1173
  - MESSAGEFORMAT 1207
  - MIRRORLOGDIR 1208
  - modifying the file 1169
  - MOVEBATCHSIZE 1209
  - MOVESIZETHRESH 1210
  - MSGINTERVAL 1211
  - NDMPCONTROLPORT 1212
  - NDMPPORTRANGE 1213
  - NDMPPREFDATAINTERFACE 1214
  - new xxix
  - NOPREEMPT 1215
  - NORETRIEVEDATE 1216
  - NUMOPENVOLSALLOWED 1217
  - QUERYAUTH 1219
  - querying 1169
  - RECLAIMDELAY 1220
  - RECLAIMPERIOD 1221
  - REPORTRETRIEVE 1222
  - REQSYSAUTHOUTFILE 1223
  - RESOURCETIMEOUT 1224
  - RESTOREINTERVAL 1225
  - RETENTIONEXTENSION 1226
  - SANDISCOVERYTIMEOUT xx, 1228
  - SANREFRESHTIME 1229
  - SEARCHMPQUEUE 1230
  - SHMPORT 1231
  - SHREDDING 1232
  - SNMPHEARTBEATINTERVAL 1233
  - SNMPMESSAGECATEGORY 1234
  - SNMPSUBAGENT 1235
  - SNMPSUBAGENTHOST 1236
  - SNMPSUBAGENTPORT 1237
  - SSLTCPADMINPORT 1238
  - SSLTCPPOINT 1239
  - tailoring 1169
  - TCPADMINPORT 1240
  - TCPBUFSIZE 1241
  - TCPNODELAY 1242
  - TCPPOINT 1243
  - TCPWINDOWSIZE 1244
  - TECBEGINEVENTLOGGING 1245
  - TECHOST 1246
  - TECPPOINT 1247

- server options (*continued*)
  - TECUTF8EVENT 1248
  - THROUGHPUTDATATHRESHOLD 1249
  - THROUGHPUTTIMETHRESHOLD 1250
  - TXNGROUPMAX 1251
  - UNIQUETDPTCEVENTS 1252
  - UNIQUETECEVENTS 1253
  - updated xxxi
  - VOLUMEHISTORY 1256
- server scripts
  - copying 103
  - defining 281
  - deleting 390
  - querying 765
  - renaming 857
  - running 887
  - running commands in parallel 577
  - running commands serially 897
  - updating 1110
- server utilities
  - deleted xxxiv
  - new xxix
  - updated xxxi
- server-to-server communications 283
  - COMMMETHOD option 1189
  - shared memory between server and client 1189
- session
  - maximum number scheduled 936
  - querying 774
- SET ACCOUNTING command 900, 952
- SET ACTLOGRETENTION command 901
- SET AUTHENTICATION command 905
- SET CLIENTACTDURATION command 906
- SET CONFIGMANAGER command 907
- SET CONFIGREFRESH command 908
- SET CROSSDEFINE command 910
- SET DBRECOVERY command 911
- SET DBREPORTMODE command 912
- SET DRMACTIVEDATASTGPOOL command 913
- SET DRMCHECKLABEL command 915
- SET DRMCMDFILENAME command 916
- SET DRMCOPYSTGPOOL command 917
- SET DRMCOURIERNAME command 918
- SET DRMDBBACKUPEXPIREDAYS command 919
- SET DRMFILEPROCESS command 921
- SET DRMINSTRPREFIX command 922
- SET DRMNOTMOUNTABLENAME command 924
- SET DRMPLANPREFIX command 925
- SET DRMPLANVPOSTFIX command 927
- SET DRMPRIMSTGPOOL command 928
- SET DRMRPFEXPIREDAYS command 929
- SET DRMVaultNAME command 931
- SET EVENTRETENTION command 932
- SET INVALIDPWLIMIT command 933
- SET LICENSEAUDITPERIOD command 934
- SET MAXCMDRETRIES command 935
- SET MAXSCHEDSESSIONS command 936
- SET MINPWLENGTH command 937
- SET PASSEXP command 938
- SET QUERYSCHEDPERIOD 940
- SET RANDOMIZE command 941
- SET REGISTRATION command 943
- SET RETRYPERIOD command 945
- SET SCHEDMODES command 946
- SET SERVERHLADDRESS command 948
- SET SERVERLLADDRESS command 949
- SET SERVERNAME command 950
- SET SERVERPASSWORD command 951
- SET SUBFILE command 952
- SET SUMMARYRETENTION command 953
- SET TAPEALERTMSG command 954
- SET TOCLOADRETENTION command 955
- SETARCHIVERETENTIONPROTECTION command 903
- SETOPT command 956
- setting
  - accounting record 900
  - authentication 905
  - configuration manager 907
  - configuration refresh 908
  - cross define of a server 910
  - frequency for client-polling 940
  - frequency for one-time client action 906
  - high level address of a server 948
  - license audit period 934
  - low level address of a server 949
  - maximum retries of a command 935
  - maximum scheduled sessions 936
  - password for a server 951
  - password for expiration date 938
  - randomization of start times 941
  - registration 943
  - retention period for activity log 901
  - retention period for event records 932
  - retry period 945
  - scheduling mode 946
  - server name 950
  - summary retention days 953
- setting communication options 1170
- shared library type 227, 1062
- SHMPORT option 1231
  - specifying a TCP/IP port address 1231
  - using shared memory 1231
- SHRED DATA command 958
- shredding data, storage pools
  - backing up 56
  - defining 300
  - moving data 541
  - updating 1126
- SHREDDING option 1232
- simultaneous write to copy storage pools
  - defining random access storage pools 299
  - defining sequential access storage pools 313
  - updating random access storage pools 1125
  - updating sequential access storage pools 1138
- single-instance store (data deduplication) 315, 324, 331, 1139, 1140, 1146, 1152
- SKIPPARTIAL parameter, AUDIT VOLUME 39
- SnapLock for managing WORM FILE volumes
  - when defining active-data pools 328
  - when defining copy storage pools 321
  - when defining sequential access storage pools 308
- SNMPHEARTBEATINTERVAL option 1233
- SNMPMESSAGECATEGORY option 1234
- SNMPSUBAGENT option 1235
- SNMPSUBAGENTHOST option 1236
- SNMPSUBAGENTPORT option 1237
- Software Support
  - contact xv
  - describing problem for IBM Software Support xvi
  - determining business impact for IBM Software Support xvi
- Sony Blue Laser RW and WORM media 198, 1040
- space management
  - DEFINE MGMTCLASS 235

- space management (*continued*)
  - description of 235
  - UPDATE MGMTCLASS 1068
- space trigger commands
  - DEFINE SPACETRIGGER 289
  - DELETE SPACETRIGGER 393
  - QUERY SPACETRIGGER 780
  - UPDATE SPACETRIGGER 1117
- SPACEMGTECHNIQUE parameter
  - DEFINE MGMTCLASS 236
  - UPDATE MGMTCLASS 1068
- SQL
  - changes to syntax in V6.1 xxxvii
  - SELECT command xxxvii
- SSLTCPADMINPORT server option 1238
- SSLTCPPOST server option 1239
- start time
  - information about 260
  - setting 941
- STARTDATE parameter
  - DEFINE SCHEDULE, administrative command 276
  - DEFINE SCHEDULE, client 268
  - UPDATE SCHEDULE, administrative command 1106
  - UPDATE SCHEDULE, client 1097
- STARTTIME parameter
  - DEFINE SCHEDULE, administrative command 276
  - DEFINE SCHEDULE, client 268
  - UPDATE SCHEDULE, administrative command 1106
  - UPDATE SCHEDULE, client 1097
- status
  - event completion 665
  - information about 782
  - querying 782
- STGPOOL parameter, MOVE DATA 541
- STGPOOLS parameter
  - GRANT AUTHORITY 490
  - REVOKE AUTHORITY 883
- storage area network (SAN)
  - DEFINE LIBRARY command 222
  - UPDATE LIBRARY command 1059
- storage pool
  - collocating
    - active-data pool 326, 1148
    - copy sequential access 318, 1142
    - primary sequential access 309, 1134
  - defining 292
  - deleting 394
  - migrating 536
  - querying 788
  - reclaiming 826
  - restoring 871
  - updating 1119
- storage pool volume
  - auditing 37
  - defining 337
  - deleting 403
  - querying 614, 818
  - scratch, active-data pool 330, 1150
  - scratch, copy sequential access 322, 1144
  - scratch, primary sequential access 310, 1135
  - updating 1157
  - varying 1166
- storage pools 289
- storage volumes
  - naming 12
  - performance increase with raw partitions 292
- support information xiii

- SWAP parameter, CHECKIN LIBVOLUME 75
- syntax diagram
  - abbreviations 8
  - default value 9
  - fragments 11
  - optional choice 8
  - repeatable choice 10
  - repeating values 10
  - required parameters 8
  - symbols 9
  - using 8
  - variables 9
- SYSCONFIG command (on NAS file server) 240, 1081
- system privilege class
  - administrative commands 18

## T

- tape
  - AUDIT LIBRARY command 34
  - AUDIT VOLUME command 37
  - CHECKIN LIBVOLUME command 70
  - CHECKOUT LIBVOLUME command 77
  - DEFINE DEVCLASS command 139
  - DEFINE LIBRARY command 222
  - DEFINE STGPOOL command 292, 302, 317
  - DEFINE VOLUME command 337
  - DELETE DEVCLASS command 359
  - DELETE DRIVE command 361
  - DELETE LIBRARY command 371
  - DELETE VOLUME command 403
  - LABEL LIBVOLUME command 523
  - QUERY DEVCLASS command 633
  - QUERY DRIVE command 641
  - QUERY LIBRARY command 686
  - QUERY LIBVOLUME command 689
  - QUERY VOLUME command 818
  - storage area network (SAN) 226, 228
  - UPDATE DEVCLASS command 986
  - UPDATE LIBRARY command 1059
  - UPDATE LIBVOLUME command 1064
  - UPDATE VOLUME command 1157
- tape-device encryption
  - 3592 151, 997
  - LTO 192, 1034
- TCPADMINPORT server option 1240
- TCPBUFSIZE option 1241
- TCPNODELAY option 1242
- TCPPOST option 1243
- TCPWINDOWSIZE option 1244
- TECBEGINEVENTLOGGING option 1245
- TECHOST option 1246
- TECPOST option 1247
- TECUTF8EVENT option 1248
- threshold, migration
  - random-access storage pools
    - high threshold 297, 1122
    - low threshold 297, 1123
  - sequential-access storage pools
    - high threshold 306, 1132
    - low threshold 306, 1132
- THROUGHPUTDATATHRESHOLD option 1249
- THROUGHPUTTIMETHRESHOLD option 1250
- time interval, setting for checking in volumes 1015, 1034, 1040
- Tivoli technical training xiv
- TOCDestination parameter 130, 980

- TOSERVER parameter
  - EXPORT ADMIN 433
  - EXPORT NODE 451
  - EXPORT POLICY 457
  - EXPORT SERVER 472
- training, Tivoli technical xiv
- transfer data by data mover 136
- troubleshooting LAN-free data movement 1162
- tsmdlst utility xx
- TXNGROUPMAX option 1251
- type, device
  - 3570 140, 987
  - 3590 143, 990
  - 3592 147, 994
  - 4MM 152, 998
  - 8MM 156, 1002
  - CARTRIDGE 162, 1007
  - CENTERA 166, 1010
  - DLT 168, 1012, 1017
  - DTF 174
  - ECARTRIDGE 177, 1020
  - FILE 182, 208, 1025, 1048
  - GENERICTAPE 185, 1029
  - LTO 188, 1031
  - OPTICAL 196, 1038
  - QIC 200, 1042
  - REMOVABLEFILE 205, 1046
  - SERVER 208, 1048
  - VOLSAFE 210, 1050
  - WORM 196, 1038
  - WORM12 196, 1038
- typographic conventions xvii

## U

- UNIQUEDPTECEVENTS option 1252
- UNIQUETECEVENTS option 1253
- UNLOCK ADMIN command 962
- UNLOCK NODE command 963
- UNLOCK PROFILE command 964
- unlocking
  - administrator 962
  - node 963
  - profile 964
- UPDATE ADMIN command 966
- UPDATE BACKUPSET 968
- UPDATE CLIENTOPT command 973
- UPDATE CLOPTSET command 974
- UPDATE COLLOGROUP command 975
- UPDATE COPYGROUP command 976
- UPDATE DATAMOVER command 984
- UPDATE DEVCLASS command 986
- UPDATE DOMAIN command 1053
- UPDATE DRIVE command 1055
- UPDATE LIBRARY command 1059
- UPDATE LIBVOLUME command 1064
- UPDATE MACHINE command 1066
- UPDATE MGMTCLASS command 1068
- UPDATE NODE command 1071
- UPDATE PATH command 1080
- UPDATE POLICYSET command 1085
- UPDATE PROFILE command 1087
- UPDATE RECOVERYMEDIA command 1088
- UPDATE SCHEDULE command 1090
- UPDATE SCRIPT command 1110
- UPDATE SERVER command 1112
- UPDATE SERVERGROUP command 1116

- UPDATE SPACETRIGGER command 1117
- UPDATE STGPOOL command 1119
- UPDATE VIRTUALFSMAPPING command 1153
- UPDATE VOLHISTORY command 1155
- UPDATE VOLUME command 1157
- updating
  - administrator 966
  - archive copy group 981
  - backup copy group 977
  - backup set 968
  - copy group 976
  - device class 986
  - domain 1053
  - drive 1055
  - library 1059
  - library volume 1064
  - management class 1068
  - node 1071
  - node group 1079
  - policy set 1085
  - profile 1087
  - retention period for a backup set 968
  - schedule 1090
  - script 1110
  - server group 1116
  - storage pool 1119
  - storage pool space triggers 1117
  - storage pool volume 1157
  - TOCDestination copy group parameter 980
  - volume history 1155
- USEREXIT option 1254
- using
  - administrative client options 5
  - command-line interface 1
  - continuation characters 11
  - macros 534
  - redirection 4
  - redirection characters 4
  - syntax diagram 8
  - system date on the server 28
- utilities
  - changes in V6 xxix
  - deleted xxxiv
  - new xxix
  - updated xxxi

## V

- VALIDATE LANFREE command 1162
- VALIDATE POLICYSET command 1164
- validating
  - LAN-free 1162
  - policy set 1164
- VARY command 1166
- varying volumes 1166
- VERDELETED parameter
  - DEFINE COPYGROUP, backup 128
  - UPDATE COPYGROUP, backup 978
- VEREXISTS parameter
  - DEFINE COPYGROUP, backup 128
  - UPDATE COPYGROUP, backup 978
- verify automated library inventory 34
- VOLSAFE
  - device type 210, 1050
  - labeling volumes 527
- volume
  - labeling library 523

- volume history
  - deleting 398
  - querying 811
- VOLUMEHISTORY option 1256
- volumes, storage
  - defining 337
  - limiting offsite volumes to be reclaimed
    - using the DEFINE STGPOOL command 321, 329
    - using the UPDATE STGPOOL command 1144, 1150
  - naming 12
  - reclaiming
    - from active-data pool 327, 1149
    - from copy sequential access 319, 1143
    - from primary sequential access 307, 1132
  - restoring 875
  - scratch, maximum in active-data pool 330, 1150
  - scratch, maximum in copy sequential access 322, 1144
  - scratch, maximum in primary sequential access 310, 1135

## W

- WAITTIME parameter
  - CHECKIN LIBVOLUME command 75
  - LABEL LIBVOLUME command 527
- WHEREACCESS parameter, UPDATE VOLUME 1159
- WHERESTATUS parameter, UPDATE VOLUME 1159
- WHERESTGPOOL parameter, UPDATE VOLUME 1159
- wildcard
  - match-any examples 13
  - match-exactly-one examples 13
  - using 13
- WORM devices and media
  - 8mm devices 157
  - DLT 169
  - IBM 3592 148
  - LTO 189
  - SnapLock for managing WORM FILE volumes 308, 321, 328
  - Sony AIT50/AIT100, supported in 8mm class
    - definition 157
  - Sony Blue Laser 198
  - StorageTek VolSafe 178
- WORM optical devices
  - defining the class 196
  - updating the class 1038
  - WORM12 197
  - WORM14 197







Program Number: 5608-E01, 5608-E02, 5608-E03, 5608-E07, 5608-E12

Printed in USA

SC23-9775-01

