

IBM Tivoli Storage Manager  
for HP-UX  
Version 7.1.6

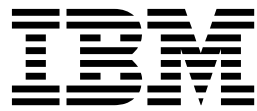
*Administrator's Reference*





IBM Tivoli Storage Manager  
for HP-UX  
Version 7.1.6

*Administrator's Reference*



**Note:**

Before you use this information and the product it supports, read the information in “Notices” on page 1619.

**Sixth edition (June 2016)**

This edition applies to version 7, release 1, modification 6 of IBM Tivoli Storage Manager (product numbers 5608-E01, 5608-E02, 5608-E03), and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1993, 2016.

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

---

# Contents

## About this publication . . . . . xi

Who should read this guide . . . . .	xi
Publications . . . . .	xi
Conventions used in this publication . . . . .	xi
What's new in Version 7.1.6 . . . . .	xii

## Chapter 1. Managing 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 . . . . .	3
Processing individual commands 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 Operations Center . . . . .	8
Issuing commands from the server console . . . . .	8
Entering administrative commands . . . . .	8
Reading syntax diagrams . . . . .	9
Using continuation characters to enter long commands . . . . .	13
Naming Tivoli Storage Manager objects . . . . .	13
Using wildcard characters to specify object names . . . . .	14
Specifying descriptions in keyword parameters . . . . .	15
Controlling command processing . . . . .	16
Server command processing . . . . .	16
Stopping background processes . . . . .	17
Performing tasks concurrently on multiple servers . . . . .	17
Routing commands to a single server . . . . .	18
Routing commands to multiple servers . . . . .	18
Routing commands to a server group . . . . .	18
Routing commands to server groups . . . . .	19
Routing commands to two servers and a server group . . . . .	19
Routing commands inside scripts . . . . .	19
Privilege classes for commands . . . . .	20
Commands requiring system privilege . . . . .	20
Commands requiring policy privilege . . . . .	23
Commands requiring storage privilege . . . . .	24
Commands requiring operator privilege . . . . .	25
Commands any administrator can issue . . . . .	26

## Chapter 2. Administrative commands 29

ACCEPT DATE (Accepts the current system date) . . . . .	30
ACTIVATE POLICYSET (Activate a new policy set) . . . . .	31
ASSIGN DEFMGMTCLASS (Assign a default management class) . . . . .	33
AUDIT commands . . . . .	35

AUDIT LDAPDIRECTORY (Audit an LDAP directory server) . . . . .	36
AUDIT LIBRARY (Audit volume inventories in an automated library) . . . . .	38
AUDIT LIBVOLUME (Verify database information for a tape volume) . . . . .	40
AUDIT LICENSES (Audit server storage usage) . . . . .	42
AUDIT VOLUME (Verify database information for a storage pool volume) . . . . .	43
BACKUP commands . . . . .	49
BACKUP DB (Back up the database) . . . . .	50
BACKUP DEVCONFIG (Create backup copies of device configuration information) . . . . .	55
BACKUP NODE (Back up a NAS node) . . . . .	57
BACKUP STGPOOL (Back up primary storage pool data to a copy storage pool) . . . . .	62
BACKUP VOLHISTORY (Save sequential volume history information) . . . . .	66
BEGIN EVENTLOGGING (Begin logging events) . . . . .	68
CANCEL commands . . . . .	70
CANCEL EXPIRATION (Cancel an expiration process) . . . . .	71
CANCEL EXPORT (Delete a suspended export operation) . . . . .	72
CANCEL PROCESS (Cancel an administrative process) . . . . .	73
CANCEL REPLICATION (Cancel node replication processes) . . . . .	75
CANCEL REQUEST (Cancel one or more mount requests) . . . . .	76
CANCEL RESTORE (Cancel a restartable restore session) . . . . .	77
CANCEL SESSION (Cancel one or more client sessions) . . . . .	78
CHECKIN LIBVOLUME (Check a storage volume into a library) . . . . .	79
CHECKOUT LIBVOLUME (Check a storage volume out of a library) . . . . .	86
CLEAN DRIVE (Clean a drive) . . . . .	92
COMMIT (Control committing of commands in a macro) . . . . .	93
COPY commands . . . . .	94
COPY ACTIVE DATA (Copy active backup data from a primary storage pool to an active-data pool) . . . . .	95
COPY CLOPTSET (Copy a client option set) . . . . .	99
COPY DOMAIN (Copy a policy domain) . . . . .	100
COPY MGMTCLASS (Copy a management class) . . . . .	102
COPY POLICYSET (Copy a policy set) . . . . .	104
COPY PROFILE (Copy a profile) . . . . .	106
COPY SCHEDULE (Copy a client or an administrative command schedule) . . . . .	108
COPY SCRIPT (Copy a Tivoli Storage Manager script) . . . . .	112
COPY SERVERGROUP (Copy a server group) . . . . .	113

DEACTIVATE DATA (Deactivate data for a client node) . . . . .	114	DEFINE SERVERGROUP (Define a server group). . . . .	322
DECOMMISSION commands . . . . .	117	DEFINE SPACETRIGGER (Define the space trigger) . . . . .	323
DECOMMISSION NODE (Decommission an application or system) . . . . .	118	DEFINE STATUSTHRESHOLD (Define a status monitoring threshold) . . . . .	326
DECOMMISSION VM (Decommission a virtual machine) . . . . .	121	DEFINE STGPOOL (Define a storage pool) . . . . .	330
DEFINE commands . . . . .	123	DEFINE SUBSCRIPTION (Define a profile subscription) . . . . .	377
DEFINE ALERTTRIGGER (Define an alert trigger) . . . . .	124	DEFINE VIRTUALFSMAPPING (Define a virtual file space mapping) . . . . .	379
DEFINE ASSOCIATION (Associate client nodes with a schedule) . . . . .	127	DEFINE VOLUME (Define a volume in a storage pool) . . . . .	381
DEFINE BACKUPSET (Define a backup set) . . . . .	129	DELETE commands . . . . .	387
DEFINE CLIENTACTION (Define a one-time client action) . . . . .	133	DELETE ALERTTRIGGER (Remove a message from an alert trigger) . . . . .	389
DEFINE CLIENTOPT (Define an option to an option set) . . . . .	139	DELETE ASSOCIATION (Delete the node association to a schedule) . . . . .	390
DEFINE CLOPTSET (Define a client option set name) . . . . .	141	DELETE BACKUPSET (Delete a backup set) . . . . .	392
DEFINE COLLOGROUP (Define a collocation group). . . . .	142	DELETE CLIENTOPT (Delete an option in an option set) . . . . .	397
DEFINE COLLOCMEMBER (Define collocation group member). . . . .	144	DELETE CLOPTSET (Delete a client option set) . . . . .	398
DEFINE COPYGROUP (Define a copy group) . . . . .	148	DELETE COLLOGROUP (Delete a collocation group). . . . .	399
DEFINE DATAMOVER (Define a data mover) . . . . .	158	DELETE COLLOCMEMBER (Delete collocation group member). . . . .	400
DEFINE DEVCLASS (Define a device class) . . . . .	161	DELETE COPYGROUP (Delete a backup or archive copy group) . . . . .	403
DEFINE DOMAIN (Define a new policy domain) . . . . .	223	DELETE DATAMOVER (Delete a data mover) . . . . .	405
DEFINE DRIVE (Define a drive to a library) . . . . .	225	DELETE DEVCLASS (Delete a device class) . . . . .	406
DEFINE EVENTSERVER (Define a server as the event server) . . . . .	230	DELETE DOMAIN (Delete a policy domain) . . . . .	407
DEFINE GRPMEMBER (Add a server to a server group) . . . . .	231	DELETE DRIVE (Delete a drive from a library) . . . . .	408
DEFINE LIBRARY (Define a library). . . . .	232	DELETE EVENT (Delete event records). . . . .	409
DEFINE MACHINE (Define machine information for disaster recovery) . . . . .	253	DELETE EVENTSERVER (Delete the definition of the event server) . . . . .	411
DEFINE MACHNODEASSOCIATION (Associate a node with a machine) . . . . .	255	DELETE FILESPACE (Delete client node data from the server) . . . . .	412
DEFINE MGMTCLASS (Define a management class) . . . . .	257	DELETE GRPMEMBER (Delete a server from a server group) . . . . .	416
DEFINE NODEGROUP (Define a node group) . . . . .	260	DELETE KEYRING (Delete password information in the key database) . . . . .	417
DEFINE NODEGROUPMEMBER (Define node group member). . . . .	261	DELETE LIBRARY (Delete a library). . . . .	418
DEFINE PATH (Define a path) . . . . .	262	DELETE MACHINE (Delete machine information). . . . .	419
DEFINE POLICYSET (Define a policy set). . . . .	271	DELETE MACHNODEASSOCIATION (Delete association between a machine and a node) . . . . .	420
DEFINE PROFASSOCIATION (Define a profile association) . . . . .	273	DELETE MGMTCLASS (Delete a management class) . . . . .	421
DEFINE PROFILE (Define a profile). . . . .	279	DELETE NODEGROUP (Delete a node group) . . . . .	422
DEFINE RECMEDMACHASSOCIATION (Associate recovery media with a machine) . . . . .	281	DELETE NODEGROUPMEMBER (Delete node group member). . . . .	423
DEFINE RECOVERYMEDIA (Define recovery media) . . . . .	283	DELETE PATH (Delete a path) . . . . .	424
DEFINE SCHEDULE (Define a client or an administrative command schedule) . . . . .	285	DELETE POLICYSET (Delete a policy set). . . . .	426
DEFINE SCRATCHPADENTRY (Define a scratch pad entry). . . . .	310	DELETE PROFASSOCIATION (Delete a profile association) . . . . .	427
DEFINE SCRIPT (Define a Tivoli Storage Manager script) . . . . .	312	DELETE PROFILE (Delete a profile). . . . .	430
DEFINE SERVER (Define a server for server-to-server communications). . . . .	315	DELETE RECMEDMACHASSOCIATION (Delete recovery media and machine association) . . . . .	432
		DELETE RECOVERYMEDIA (Delete recovery media) . . . . .	433

DELETE SCHEDULE (Delete a client or an administrative command schedule) . . . . .	434	GRANT AUTHORITY (Add administrator authority) . . . . .	554
DELETE SCRATCHPADENTRY (Delete a scratch pad entry) . . . . .	437	GRANT PROXYNODE (Grant proxy authority to a client node) . . . . .	558
DELETE SCRIPT (Delete command lines from a script or delete the entire script) . . . . .	438	HALT (Shut down the server) . . . . .	559
DELETE SERVER (Delete a server definition)	439	HELP (Get help on commands and error messages)	561
DELETE SERVERGROUP (Delete a server group) . . . . .	440	IDENTIFY DUPLICATES (Identify duplicate data in a storage pool) . . . . .	563
DELETE SPACETRIGGER (Delete the storage pool space triggers) . . . . .	441	IMPORT commands . . . . .	567
DELETE STATUSTHRESHOLD (Delete a status monitoring threshold) . . . . .	442	IMPORT ADMIN (Import administrator information) . . . . .	568
DELETE STGPOOL (Delete a storage pool) . . . . .	444	IMPORT NODE (Import client node information) . . . . .	572
DELETE SUBSCRIBER (Delete subscriptions from a configuration manager database) . . . . .	446	IMPORT POLICY (Import policy information)	579
DELETE SUBSCRIPTION (Delete a profile subscription) . . . . .	447	IMPORT SERVER (Import server information)	582
DELETE VIRTUALFSMAPPING (Delete a virtual file space mapping) . . . . .	448	INSERT MACHINE (Insert machine characteristics information or recovery instructions) . . . . .	588
DELETE VOLHISTORY (Delete sequential volume history information) . . . . .	449	ISSUE MESSAGE (Issue a message from a server script) . . . . .	589
DELETE VOLUME (Delete a storage pool volume) . . . . .	454	LABEL LIBVOLUME (Label a library volume) . . . . .	591
DISABLE commands . . . . .	457	LOAD DEFALERTTRIGGERS (Load the default set of alert triggers) . . . . .	598
DISABLE EVENTS (Disable events for event logging) . . . . .	458	LOCK commands . . . . .	600
DISABLE REPLICATION (Prevent outbound replication processing on a server) . . . . .	462	LOCK ADMIN (Lock out an administrator) . . . . .	601
DISABLE SESSIONS (Prevent new sessions from accessing Tivoli Storage Manager) . . . . .	463	LOCK NODE (Lock out a client node) . . . . .	603
DISMOUNT command . . . . .	465	LOCK PROFILE (Lock a profile) . . . . .	605
DISMOUNT VOLUME (Dismount a volume by volume name) . . . . .	466	MACRO (Invoke a macro) . . . . .	607
DISPLAY OBJNAME (Display a full object name)	467	MIGRATE STGPOOL (Migrate storage pool to next storage pool) . . . . .	609
ENABLE commands . . . . .	468	MOVE commands . . . . .	612
ENABLE EVENTS (Enable server or client events for logging) . . . . .	469	MOVE DATA (Move files on a storage pool volume) . . . . .	613
ENABLE REPLICATION (Allow outbound replication processing on a server) . . . . .	472	MOVE DRMEDIA (Move disaster recovery media offsite and back onsite) . . . . .	617
ENABLE SESSIONS (Resume user activity on the server) . . . . .	473	MOVE GRPMEMBER (Move a server group member) . . . . .	633
END EVENTLOGGING (Stop logging events) . . . . .	475	MOVE MEDIA (Move sequential-access storage pool media) . . . . .	634
EXPIRE INVENTORY (Manually start inventory expiration processing) . . . . .	477	MOVE NODEDATA (Move data by node in a sequential access storage pool) . . . . .	642
EXPORT commands . . . . .	482	NOTIFY SUBSCRIBERS (Notify managed servers to update profiles) . . . . .	652
EXPORT ADMIN (Export administrator information) . . . . .	483	PERFORM LIBACTION (Define or delete all drives and paths for a library) . . . . .	653
EXPORT NODE (Export client node information) . . . . .	490	PING SERVER (Test the connection between servers) . . . . .	657
EXPORT POLICY (Export policy information)	512	PREPARE (Create a recovery plan file) . . . . .	658
EXPORT SERVER (Export server information)	519	QUERY commands . . . . .	664
EXTEND DBSPACE (Increase space for the database) . . . . .	538	QUERY ACTLOG (Query the activity log) . . . . .	667
GENERATE commands . . . . .	541	QUERY ADMIN (Display administrator information) . . . . .	673
GENERATE BACKUPSET (Generate a backup set of Backup-Archive Client data) . . . . .	542	QUERY ALERTTRIGGER (Query the list of defined alert triggers) . . . . .	678
GENERATE BACKUPSETTOC (Generate a table of contents for a backup set) . . . . .	551	QUERY ALERTSTATUS (Query the status of an alert) . . . . .	680
GRANT commands . . . . .	553	QUERY ASSOCIATION (Query client node associations with a schedule) . . . . .	685
		QUERY AUDITOCCUPANCY (Query client node storage utilization) . . . . .	687
		QUERY BACKUPSET (Query a backup set) . . . . .	690

QUERY BACKUPSETCONTENTS (Query contents of a backup set) . . . . .	695	QUERY OPTION (Query server options) . . . . .	838
QUERY CLOPTSET (Query a client option set) . . . . .	698	QUERY PATH (Display a path definition) . . . . .	840
QUERY COLLOCGROUP (Query a collocation group). . . . .	700	QUERY POLICYSET (Query a policy set) . . . . .	844
QUERY CONTENT (Query the contents of a storage pool volume). . . . .	703	QUERY PROCESS (Query one or more server processes) . . . . .	847
QUERY COPYGROUP (Query copy groups) . . . . .	710	QUERY PROFILE (Query a profile) . . . . .	851
QUERY DATAMOVER (Display data mover definitions) . . . . .	715	QUERY PROXYNODE (Query proxy authority for a client node) . . . . .	854
QUERY DB (Display database information) . . . . .	718	QUERY PVUESTIMATE (Display processor value unit estimate) . . . . .	855
QUERY DBSPACE (Display database storage space) . . . . .	721	QUERY RECOVERYMEDIA (Query recovery media) . . . . .	859
QUERY DEVCLASS (Display information on one or more device classes). . . . .	723	QUERY REPLICATION (Query node replication processes) . . . . .	862
QUERY DIRSPACE (Query storage utilization of FILE directories) . . . . .	727	QUERY REPLNODE (Display information about replication status for a client node) . . . . .	875
QUERY DOMAIN (Query a policy domain) . . . . .	728	QUERY REPLRULE (Query replication rules) . . . . .	879
QUERY DRIVE (Query information about a drive) . . . . .	731	QUERY REPLSERVER (Query a replication server). . . . .	881
QUERY DRMEDIA (Query disaster recovery media) . . . . .	735	QUERY REQUEST (Query one or more pending mount requests) . . . . .	883
QUERY DRMSTATUS (Query disaster recovery manager system parameters) . . . . .	744	QUERY RESTORE (Query restartable restore sessions) . . . . .	884
QUERY ENABLED (Query enabled events) . . . . .	747	QUERY RPFCONTENT (Query recovery plan file contents stored on a target server) . . . . .	887
QUERY EVENT (Query scheduled and completed events). . . . .	749	QUERY RPFFILE (Query recovery plan file information stored on a target server) . . . . .	889
QUERY EVENTRULES (Query rules for server or client events) . . . . .	761	QUERY SCHEDULE (Query schedules). . . . .	892
QUERY EVENTSERVER (Query the event server). . . . .	764	QUERY SCRATCHPADENTRY (Query a scratch pad entry) . . . . .	900
QUERY EXPORT (Query for active or suspended export operations) . . . . .	765	QUERY SCRIPT (Query Tivoli Storage Manager scripts) . . . . .	902
QUERY FILESPACE (Query one or more file spaces) . . . . .	772	QUERY SERVER (Query a server) . . . . .	905
QUERY LIBRARY (Query a library) . . . . .	780	QUERY SERVERGROUP (Query a server group) . . . . .	910
QUERY LIBVOLUME (Query a library volume) . . . . .	783	QUERY SESSION (Query client sessions) . . . . .	912
QUERY LICENSE (Display license information) . . . . .	786	QUERY SHREDSTATUS (Query shredding status) . . . . .	917
QUERY LOG (Display information about the recovery log) . . . . .	790	QUERY SPACETRIGGER (Query the space triggers) . . . . .	919
QUERY MACHINE (Query machine information). . . . .	792	QUERY SSLKEYRINGPW (Query SSL key database file password) . . . . .	921
QUERY MEDIA (Query sequential-access storage pool media) . . . . .	795	QUERY STATUS (Query system parameters) . . . . .	922
QUERY MGMTCLASS (Query a management class) . . . . .	801	QUERY STATUSTHRESHOLD (Query status monitoring thresholds) . . . . .	932
QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status) . . . . .	804	QUERY STGPOOL (Query storage pools) . . . . .	935
QUERY MONITORSTATUS (Query the monitoring status). . . . .	807	QUERY SUBSCRIBER (Display subscriber information). . . . .	948
QUERY MOUNT (Display information on mounted sequential access volumes). . . . .	811	QUERY SUBSCRIPTION (Display subscription information). . . . .	950
QUERY NASBACKUP (Query NAS backup images) . . . . .	813	QUERY SYSTEM (Query the system configuration and capacity). . . . .	952
QUERY NODE (Query nodes). . . . .	817	QUERY TAPEALERTMSG (Display status of SET TAPEALERTMSG command) . . . . .	954
QUERY NODEDATA (Query client data in volumes) . . . . .	829	QUERY TOC (Display table of contents for a backup image) . . . . .	955
QUERY NODEGROUP (Query a node group) . . . . .	832	QUERY VIRTUALFSMAPPING (Query a virtual file space mapping) . . . . .	958
QUERY OCCUPANCY (Query client file spaces in storage pools) . . . . .	834	QUERY VOLHISTORY (Display sequential volume history information) . . . . .	960
		QUERY VOLUME (Query storage pool volumes) . . . . .	968



QUIT (End the interactive mode of the administrative client) . . . . .	975	SET ALERTCLOSEDDURATION (Set the duration of a closed alert) . . . . .	1084
RECLAIM STGPOOL (Reclaim volumes in a sequential-access storage pool) . . . . .	976	SET ALERTEMAIL (Set the alert monitor to email alerts to administrators) . . . . .	1085
RECONCILE VOLUMES (Reconcile differences in the virtual volume definitions) . . . . .	979	SET ALERTEMAILFROMADDR (Set the email address of the sender) . . . . .	1086
REGISTER commands . . . . .	982	SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name) . . . . .	1087
REGISTER ADMIN (Register an administrator ID) . . . . .	983	SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port) . . . . .	1088
REGISTER LICENSE (Register a new license) . . . . .	987	SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email) . . . . .	1089
REGISTER NODE (Register a node) . . . . .	989	SET ALERTINACTIVEDURATION (Set the duration of an inactive alert) . . . . .	1090
REMOVE commands . . . . .	1007	SET ALERTMONITOR (Set the alert monitor to on or off) . . . . .	1091
REMOVE ADMIN (Delete an administrator user ID) . . . . .	1008	SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts) . . . . .	1092
REMOVE NODE (Delete a node or an associated machine node) . . . . .	1010	SET ARCHIVERETENTIONPROTECTION (Activate data retention protection) . . . . .	1093
REMOVE REPLNODE (Remove a client node from replication) . . . . .	1012	SET ARREPLRULEDEFAULT (Set the server replication rule for archive data) . . . . .	1095
REMOVE REPLSERVER (Remove a replication server) . . . . .	1014	SET AUTHENTICATION (Set password authentication) . . . . .	1097
RENAME commands . . . . .	1016	SET BKREPLRULEDEFAULT (Set the server replication rule for backup data) . . . . .	1098
RENAME ADMIN (Rename an administrator) . . . . .	1017	SET CLIENTACTDURATION (Set the duration period for the client action) . . . . .	1101
RENAME FILESPACE (Rename a client file space on the server) . . . . .	1019	SET CONFIGMANAGER (Specify a configuration manager) . . . . .	1102
RENAME NODE (Rename a node) . . . . .	1023	SET CONFIGREFRESH (Set managed server configuration refresh) . . . . .	1103
RENAME SCRIPT (Rename a Tivoli Storage Manager script) . . . . .	1025	SET CONTEXTMESSAGING (Set message context reporting on or off) . . . . .	1104
RENAME SERVERGROUP (Rename a server group) . . . . .	1026	SET CPUINFOREFRESH (Refresh interval for the client workstation information scan) . . . . .	1105
RENAME STGPOOL (Change the name of a storage pool) . . . . .	1027	SET CROSSDEFINE (Specifies whether to cross-define servers) . . . . .	1106
REPLICATE NODE (Replicate data in file spaces that belong to a client node) . . . . .	1028	SET DBRECOVERY (Set the device class for automatic backups) . . . . .	1107
REPLY (Allow a request to continue processing) . . . . .	1038	SET DEDUPVERIFICATIONLEVEL (Set the percentage of extents to verify) . . . . .	1109
RESET PASSEXP (Reset password expiration) . . . . .	1039	SET DEFAULTAUTHENTICATION (Set the default authentication method for <b>REGISTER NODE</b> and <b>REGISTER ADMIN</b> commands) . . . . .	1111
RESTART EXPORT (Restart a suspended export operation) . . . . .	1041	SET DISSIMILARPOLICIES (Enable the policies on the target replication server to manage replicated data) . . . . .	1113
RESTORE commands . . . . .	1043	SET DRMACTIVEDATASTGPOOL (Specify the active-data pools to be managed by DRM) . . . . .	1115
RESTORE NODE (Restore a NAS node) . . . . .	1044	SET DRMCHECKLABEL (Specify label checking) . . . . .	1116
RESTORE STGPOOL (Restore storage pool data from a copy pool or an active-data pool) . . . . .	1049	SET DRMCMDFILENAME (Specify the name of a file to contain commands) . . . . .	1117
RESTORE VOLUME (Restore primary volume data from a copy pool or an active-data pool) . . . . .	1053	SET DRMCOPYSTGPOOL (Specify the copy storage pools to be managed by DRM) . . . . .	1118
REVOKE commands . . . . .	1057	SET DRMCOURIERNAME (Specify the courier name) . . . . .	1119
REVOKE AUTHORITY (Remove administrator authority) . . . . .	1058	SET DRMDBBACKUPEXPIREDAYS (Specify DB backup series expiration) . . . . .	1120
REVOKE PROXYNODE (Revoke proxy authority for a client node) . . . . .	1062		
ROLLBACK (Rollback uncommitted changes in a macro) . . . . .	1063		
RUN (Run a Tivoli Storage Manager script) . . . . .	1064		
SELECT (Perform an SQL query of the Tivoli Storage Manager database) . . . . .	1067		
SET commands . . . . .	1077		
SET ACCOUNTING (Set accounting records on or off) . . . . .	1080		
SET ACTLOGRETENTION (Set the retention period or the size of the activity log) . . . . .	1081		
SET ALERTACTIVEDURATION (Set the duration of an active alert) . . . . .	1083		

SET DRMFILEPROCESS (Specify file processing) . . . . .	1122	SET SCRATCHPADRETENTION (Set scratch pad retention time) . . . . .	1168
SET DRMINSTRPREFIX (Specify the prefix for recovery instructions file names) . . . . .	1123	SET SERVERHLADDRESS (Set the high-level address of a server) . . . . .	1169
SET DRMNOTMOUNTABLENAME (Specify the not mountable location name) . . . . .	1125	SET SERVERLLADDRESS (Set the low-level address of a server) . . . . .	1170
SET DRMPLANPREFIX (Specify a prefix for recovery plan file names) . . . . .	1126	SET SERVERNAME (Specify the server name)	1171
SET DRMPLANVPOSTFIX (Specify replacement volume names) . . . . .	1128	SET SERVERPASSWORD (Set password for server) . . . . .	1172
SET DRMPRIMSTGPOOL (Specify the primary storage pools to be managed by DRM) . . . . .	1129	SET SPREPLRULEDEFAULT (Set the server replication rule for space-managed data) . . . . .	1173
SET DRMRPFEXPIREDAYS (Set criteria for recovery plan file expiration) . . . . .	1130	SET SSLKEYRINGPW (Set the SSL key ring password) . . . . .	1175
SET DRMVaultNAME (Specify the vault name) . . . . .	1132	SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation) . . . . .	1176
SET EVENTRETENTION (Set the retention period for event records) . . . . .	1133	SET STATUSMONITOR (Specifies whether to enable status monitoring) . . . . .	1178
SET FAILOVERHLADDRESS (Set a failover high level address) . . . . .	1134	SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring) . . . . .	1180
SET INVALIDPWLIMIT (Set the number of invalid logon attempts) . . . . .	1135	SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation) . . . . .	1182
SET LDAPPASSWORD (Set the LDAP password for the server) . . . . .	1137	SET SUBFILE (Set subfile backup for client nodes) . . . . .	1184
SET LDAPUSER (Specify the user ID for the LDAP directory server) . . . . .	1138	SET SUMMARYRETENTION (Set number of days to keep data in activity summary table) . . . . .	1185
SET LICENSEAUDITPERIOD (Set license audit period) . . . . .	1140	SET TAPEALERTMSG (Set tape alert messages on or off) . . . . .	1186
SET MAXCMDRETRIES (Set the maximum number of command retries) . . . . .	1141	SET TOCLOADRETENTION (Set load retention period for table of contents) . . . . .	1187
SET MAXSCHEDULESESSIONS (Set maximum scheduled sessions) . . . . .	1142	SET VMATRISKINTERVAL (Specifies the at-risk mode for an individual VM filespace) . . . . .	1188
SET MINPWLENGTH (Set minimum password length) . . . . .	1144	SETOPT (Set a server option for dynamic update)	1190
SET MONITOREDSEVERGROUP (Set the group of monitored servers) . . . . .	1145	SHRED DATA . . . . .	1192
SET MONITORINGADMIN (Set the name of the monitoring administrator) . . . . .	1146	SUSPEND EXPORT (Suspend a currently running export operation) . . . . .	1194
SET NODEATRISKINTERVAL (Specifies at-risk mode for an individual node) . . . . .	1147	UNLOCK commands . . . . .	1195
SET PASSEXP (Set password expiration date)	1149	UNLOCK ADMIN (Unlock an administrator)	1196
SET PRODUCTOFFERING (Set the product offering that is licensed to your enterprise) . . . . .	1151	UNLOCK NODE (Unlock a client node) . . . . .	1198
SET QUERYSCHEDPERIOD (Set query period for polling client nodes) . . . . .	1153	UNLOCK PROFILE (Unlock a profile) . . . . .	1200
SET RANDOMIZE (Set randomization of scheduled start times) . . . . .	1154	UPDATE commands . . . . .	1201
SET REGISTRATION (Set open or closed registration) . . . . .	1156	UPDATE ALERTTRIGGER (Update a defined alert trigger) . . . . .	1202
SET REPLRECOVERDAMAGED (Specify whether damaged files are recovered from a replication server) . . . . .	1158	UPDATE ALERTSTATUS (Update the status of an alert) . . . . .	1205
SET REPLRETENTION (Set the retention period for replication records) . . . . .	1161	UPDATE ADMIN (Update an administrator)	1207
SET REPLSERVER (Set the target replication server) . . . . .	1163	UPDATE BACKUPSET (Update a retention value assigned to a backup set) . . . . .	1211
SET RETRYPERIOD (Set time between retry attempts) . . . . .	1165	UPDATE CLIENTOPT (Update a client option sequence number) . . . . .	1216
SET SCHEDMODES (Select a central scheduling mode) . . . . .	1166	UPDATE CLOPTSET (Update a client option set description) . . . . .	1217
		UPDATE COLLOCGROUP (Update a collocation group) . . . . .	1218
		UPDATE COPYGROUP (Update a copy group)	1219
		UPDATE DATAMOVER (Update a data mover) . . . . .	1227
		UPDATE DEVCLASS (Update the attributes of a device class) . . . . .	1229
		UPDATE DOMAIN (Update a policy domain)	1286

UPDATE DRIVE (Update a drive) . . . . .	1288	Message options . . . . .	1462
UPDATE FILESPACE (Update file-space node-replication rules) . . . . .	1292	Event logging options . . . . .	1463
UPDATE LIBRARY (Update a library). . . . .	1297	Security options and licensing options. . . . .	1463
UPDATE LIBVOLUME (Change the status of a storage volume) . . . . .	1313	Miscellaneous options . . . . .	1464
UPDATE MACHINE (Update machine information) . . . . .	1315	3494SHARED . . . . .	1465
UPDATE MGMTCLASS (Update a management class) . . . . .	1317	ACSACCESSID . . . . .	1466
UPDATE NODE (Update node attributes) . . . . .	1320	ACSLOCKDRIVE . . . . .	1467
UPDATE NODEGROUP (Update a node group) . . . . .	1338	ACSQUICKINIT . . . . .	1468
UPDATE PATH (Change a path) . . . . .	1339	ACSTIMEOUTX . . . . .	1469
UPDATE POLICYSET (Update a policy set description) . . . . .	1347	ACTIVELOGDIRECTORY . . . . .	1470
UPDATE PROFILE (Update a profile description) . . . . .	1349	ACTIVELOGSIZE . . . . .	1471
UPDATE RECOVERYMEDIA (Update recovery media) . . . . .	1350	ADMINCOMMTIMEOUT . . . . .	1472
UPDATE REPLRULE (Update replication rules)	1352	ADMINIDLETIMEOUT . . . . .	1473
UPDATE SCHEDULE (Update a schedule)	1354	ADMINONCLIENTPORT . . . . .	1474
UPDATE SCRATCHPADENTRY (Update a scratch pad entry) . . . . .	1378	ALIASHALT . . . . .	1475
UPDATE SCRIPT (Update a Tivoli Storage Manager script) . . . . .	1379	ALLOWREORGINDEX. . . . .	1476
UPDATE SERVER (Update a server defined for server-to-server communications) . . . . .	1382	ALLOWREORGTABLE. . . . .	1477
UPDATE SERVERGROUP (Update a server group description) . . . . .	1386	ARCHFAILOVERLOGDIRECTORY. . . . .	1478
UPDATE SPACETRIGGER (Update the space triggers). . . . .	1387	ARCHLOGCOMPRESS. . . . .	1479
UPDATE STATUSTHRESHOLD (Update a status monitoring threshold) . . . . .	1389	ARCHLOGDIRECTORY . . . . .	1480
UPDATE STGPOOL (Update a storage pool)	1393	ARCHLOGUSEDTHRESHOLD . . . . .	1481
UPDATE VIRTUALFSMAPPING (Update a virtual file space mapping) . . . . .	1434	ASSISTVCRRECOVERY . . . . .	1482
UPDATE VOLHISTORY (Update sequential volume history information) . . . . .	1436	AUDITSTORAGE . . . . .	1483
UPDATE VOLUME (Change a storage pool volume). . . . .	1438	CHECKTAPEPOS . . . . .	1484
VALIDATE commands . . . . .	1442	CLIENTDEDUPTXNLIMIT . . . . .	1486
VALIDATE LANFREE (Validate LAN-Free paths) . . . . .	1443	COMMETHOD . . . . .	1488
VALIDATE POLICYSET (Verify a policy set)	1445	COMMTIMEOUT . . . . .	1490
VALIDATE REPLICATION (Validate replication for a client node) . . . . .	1447	DBDIAGLOGSIZE . . . . .	1491
VALIDATE REPLPOLICY (Verify the policies on the target replication server) . . . . .	1452	DBDIAGPATHFSTHRESHOLD . . . . .	1493
VARY (Bring a random access volume online or offline) . . . . .	1455	DBMEMPERCENT . . . . .	1494
<b>Chapter 3. Server options . . . . .</b>	<b>1457</b>	DBMTCPPORT . . . . .	1495
Modifying server options . . . . .	1457	DEDUPREQUIRESBACKUP . . . . .	1496
Types of server options. . . . .	1458	DEDUPTIER2FILESIZE. . . . .	1497
Server communication options . . . . .	1458	DEDUPTIER3FILESIZE. . . . .	1498
Server storage options . . . . .	1460	DEVCONFIG . . . . .	1499
Client-server options . . . . .	1461	DISABLEREORGTABLE . . . . .	1500
Date, number, time, and language options . . . . .	1461	DISABLESCHEDS . . . . .	1501
Database options . . . . .	1461	DISALLOWDES . . . . .	1502
Data transfer options . . . . .	1462	DISPLAYLFINFO. . . . .	1503
		DNSLOOKUP. . . . .	1504
		DRIVEACQUIRERETRY . . . . .	1505
		ENABLENASDEDUP . . . . .	1506
		EVENTSERVER . . . . .	1507
		EXPINTERVAL . . . . .	1508
		EXPQUIET . . . . .	1509
		FFDCLOGNAME . . . . .	1510
		FFDCMAXLOGSIZE. . . . .	1511
		FFDCNUMLOGS. . . . .	1512
		FILEEXIT . . . . .	1513
		FILETEXTEXIT . . . . .	1514
		FSUSEDTHRESHOLD . . . . .	1515
		IDLETIMEOUT . . . . .	1516
		KEEPALIVE . . . . .	1517
		KEEPALIVETIME . . . . .	1518
		KEEPALIVEINTERVAL. . . . .	1519
		LANGUAGE . . . . .	1520
		LDAPCACHEDURATION. . . . .	1522
		LDAPURL . . . . .	1523
		MAXSESSIONS . . . . .	1524
		MESSAGEFORMAT. . . . .	1525
		MIRRORLOGDIRECTORY . . . . .	1526

MOVEBATCHSIZE . . . . .	1527
MOVESIZETHRESH . . . . .	1528
MSGINTERVAL . . . . .	1529
NDMPCONNECTIONTIMEOUT . . . . .	1530
NDMPCONTROLPORT . . . . .	1531
NDMPENABLEKEEPALIVE . . . . .	1532
NDMPPORTRANGE . . . . .	1533
NDMPPREFDATAINTERFACE . . . . .	1534
NOPREEMPT . . . . .	1535
NORETRIEVEDATE . . . . .	1536
NUMOPENVOLSAALLOWED . . . . .	1537
QUERYAUTH . . . . .	1539
RECLAIMDELAY . . . . .	1540
RECLAIMPERIOD . . . . .	1541
REORGBEGINTIME . . . . .	1542
REORGDURATION . . . . .	1543
REPORTRETRIEVE . . . . .	1544
REPLBATCHSIZE . . . . .	1545
REPLSIZETHRESH . . . . .	1546
REQSYSAUTHOUTFILE . . . . .	1547
RESOURCE TIMEOUT . . . . .	1548
RESTOREINTERVAL . . . . .	1549
RETENTIONEXTENSION . . . . .	1550
SEARCHMPQUEUE . . . . .	1551
SERVERDEDUPTXNLIMIT . . . . .	1552
SHMPORT . . . . .	1554
SHREDDING . . . . .	1555
SNMPHEARTBEATINTERVAL . . . . .	1556
SNMPMESSAGECATEGORY . . . . .	1557
SNMPSUBAGENT . . . . .	1558
SNMPSUBAGENTHOST . . . . .	1559
SNMPSUBAGENTPORT . . . . .	1560
SSLDISABLELEGACYTLS . . . . .	1561
SSLFIPSMODE . . . . .	1562
SSLHIDELEGACYTLS . . . . .	1563
SSLINITTIMEOUT . . . . .	1564
SSLTCPADMINPORT . . . . .	1565
SSLTCPPOINT . . . . .	1566
SSLTLS12 . . . . .	1567
TCPADMINPORT . . . . .	1568
TCPBUFSIZE . . . . .	1569
TCPNODELAY . . . . .	1570
TCPPOINT . . . . .	1571
TCPWINDOWSIZE . . . . .	1572
TECBEGINEVENTLOGGING . . . . .	1573
TECHOST . . . . .	1574
TECPPOINT . . . . .	1575

TECUTF8EVENT . . . . .	1576
THROUGHPUTDATATHRESHOLD . . . . .	1577
THROUGHPUTTIMETHRESHOLD . . . . .	1578
TXNGROUPMAX . . . . .	1579
UNIQUETDPTECEVENTS . . . . .	1580
UNIQUETECEVENTS . . . . .	1581
USEREXIT . . . . .	1582
VERBCHECK . . . . .	1583
VOLUMEHISTORY . . . . .	1584

**Chapter 4. Server utilities. . . . . 1585**

DSMSERV (Start the server) . . . . .	1586
Server startup script: rc.dsmserv . . . . .	1588
DSMSERV DISPLAY DBSPACE (Display information about database storage space) . . . . .	1589
DSMSERV DISPLAY LOG (Display recovery log information) . . . . .	1590
DSMSERV EXTEND DBSPACE (Increase space for the database) . . . . .	1592
DSMSERV FORMAT (Format the database and log) . . . . .	1594
DSMSERV INSERTDB (Move a server database into an empty database) . . . . .	1597
DSMSERV LOADFORMAT (Format a database) . . . . .	1600
DSMSERV REMOVEDB (Remove a database) . . . . .	1602
DSMSERV RESTORE DB (Restore the database) . . . . .	1604
DSMSERV RESTORE DB (Restore a database to its most current state) . . . . .	1605
DSMSERV RESTORE DB (Restore a database to a point-in-time) . . . . .	1608

**Appendix A. Return codes for use in IBM Tivoli Storage Manager scripts . 1613**

**Appendix B. Accessibility features for the Tivoli Storage Manager product family . . . . . 1617**

**Notices . . . . . 1619**

**Glossary . . . . . 1623**

**Index . . . . . 1625**

---

## About this publication

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

This publication provides you with the commands and options that you can use to manage the Tivoli Storage Manager server.

---

## Who should read this guide

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

The Tivoli Storage Manager product family includes IBM Tivoli Storage FlashCopy® Manager, IBM Tivoli Storage Manager for Space Management, IBM Tivoli Storage Manager for Databases, and several other storage management products from IBM.

To view IBM product documentation, see IBM Knowledge Center.

---

## Conventions used in this publication

- Command to be entered on the HP-UX 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.

---

## What's new in Version 7.1.6

IBM Tivoli Storage Manager Version 7.1.6 introduces new features and updates.

For a list of new features and updates in this release and previous releases, see What's new.

New and changed information in this product documentation is indicated by a vertical bar (|) to the left of the change.

---

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

Tivoli Storage Manager provides several different command-line interfaces for managing Tivoli Storage Manager servers.

### About this task

The following command-line interfaces are available:

#### Administrative command-line client

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. The administrative client can be accessed remotely.

From the administrative client, you can issue any server commands.

#### Server console

The server console is a command-line window on the system where the server is installed. Therefore, to use the server console, you must be at the physical location of the server system.

Compared to the administrative client, the capabilities of the server console are limited. From the server console, you cannot issue certain commands, and you cannot route commands to other servers. Also, you cannot specify that certain commands process before other commands can be issued. However, this limitation can be useful if, for example, you want to run two commands in quick succession.

#### Operations Center command line

From the Operations Center, you can access the Tivoli Storage Manager command line. You might want to use this command line to issue server commands to complete certain Tivoli Storage Manager tasks that are not supported in the Operations Center.

Server scripts provide for automation of common administrative tasks. A macro is a file that contains one or more Tivoli Storage Manager administrative commands. When you issue the **MACRO** command, the server processes all commands in the macro file in order, including commands that are contained in any nested macros.

---

## Issuing commands from the administrative client

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

### About this task

Ensure that your administrative client and your server are running in compatible languages. See “LANGUAGE” on page 1520 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.

**Tip:** Text strings that are sent from the client to the server do not depend on the server language setting. The text is displayed properly if the administrative client runs in the same locale when sending the string and when receiving the string.

For example, assume that you update a node contact field with a value that contains national characters (update node *myNode* contact=*NLcontact\_info*), and later query the node (query node *myNode* format=detailed). If the client is running in the same locale when you update as when you query, the *NLcontact\_info* displays properly. If you update the node contact field when the client is running in one locale, and query the node when the client is running in a different locale, the *NLcontact\_info* might not display properly.

## Starting and stopping the administrative client

Use the **DSMADMC** command to start an administrative client session.

### About this task

The Tivoli Storage Manager server must be running before an administrative client can connect.

### Procedure

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

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

By entering the **DSMADMC** command with the **-ID** and **-PASSWORD** options as shown, you are not prompted for a user ID and password.

- To stop an administrative command-line client session, enter the following command:

```
quit
```

- To interrupt a **DSMADMC** command before the Tivoli Storage Manager server finishes processing it, use the UNIX `kill -9` command from an available command line. Do not press `Ctrl+C` because, while it ends the session, it can lead to unexpected results.

## 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.

### Procedure

- To start an administrative client session in console mode, enter the following command:

```
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, use a keyboard break sequence.

Operating system	Break sequence
UNIX and Linux clients	Ctrl+C
Windows clients	Ctrl+C or Ctrl+Break



## 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.

### Procedure

- To start an administrative client session in mount mode, enter the following command:  
`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, use a keyboard break sequence.

Operating system	Break sequence
UNIX and Linux clients	Ctrl+C
Windows clients	Ctrl+C or Ctrl+Break

## Processing individual commands from the administrative client

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

### Procedure

To start an administrative client session in batch mode, use the following command: `dsmadmc 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.

### About this task

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.

## Procedure

To start an administrative session in interactive mode, use the following command:  
`dsmadm`

You can use continuation characters when using interactive mode. For more information, see “Using continuation characters to enter long commands” on page 13.

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

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

```
dsmadm server_command
```

## Formatting output from commands

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

### Procedure

- 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.

### About this task

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

When redirecting output, follow the naming conventions of the operating system where you are running the administrative client.

### Procedure

The examples in the following table show how to redirect command output.

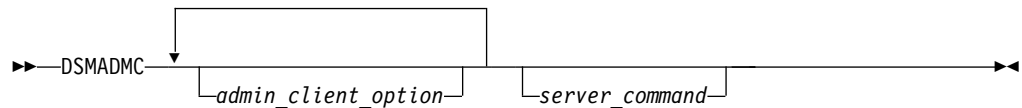
Task	Procedure
Redirect the output of a <b>QUERY DOMAIN</b> command to a new file in batch or interactive mode	Use a single greater-than sign ( <code>&gt;</code> ) to redirect the output to a new file or write over an existing file:  <code>dsmadm -id=xxx -pa=xxx query domain acctg &gt; dominfo.acc</code>

Task	Procedure
Append the output of a <b>QUERY DOMAIN</b> command to the end of an existing file in batch or interactive mode	Use two consecutive greater-than signs (>>) to append the output to the end of an existing file:  dsmdmc -id=xxx -pa=xxx query domain acctg >> dominfo.acc
Redirect all output from an administrative client session in console mode to a program called filter.exe	Use the vertical bar ( ) to direct all output for a session to a program:  dsmdmc -console -id=admin -password=xxx   filter.exe  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.
In console mode, redirect all output to a file	Specify the <b>-OUTFILE</b> option with a destination file name. For example, the following command redirects all output to the save.out file:  dsmdmc -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.

### Syntax



### Example of using administrative client options

You can enter the **DSMADMC** command with your user ID and password by using the **-ID** and **-PASSWORD** options so that you are not 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:

```
dsmdmc -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.

**-CHECKAliaShalt**

Allows the administrative client to recognize an alias for the **HALT** command as set in the **ALIASHALT** server option. See “**ALIASHALT**” on page 1475 for details.

**-COMMAdelimited**

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 you redirect 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. Most server console output is echoed to your screen. The exception are items such as responses to query commands that are issued from the console, trace output, or any system messages that displayed on the console.

**-DATAONLY=NO or YES**

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**

You can 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 after the command prompt and commands that are entered from the keyboard are displayed underneath the prompt. If this option is not specified, commands entered from the keyboard are displayed to the right side 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 that is managed by the server.

**-OUTfile**

Specifies that output from a server query is displayed in one row. If the output

in a row exceeds the column width that is defined by the server, the output is displayed on multiple lines in that row. 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 connects to. The `SERVERADDRESS` option is supported by administrative clients that are 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 you redirect 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 that are 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 that are running on Windows operating systems and is valid on the Windows administrative client command line.


In addition to the options that are 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 Operations Center

From the Operations Center command-line interface, you can issue commands to manage Tivoli Storage Manager servers that are configured as hub or spoke servers.

### Procedure

To open the command-line interface, hover over the globe icon  in the Operations Center menu bar, and click **Command Builder**.

---

## 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.

### About this task

If you have system privilege, you can revoke or grant new privileges to the `SERVER_CONSOLE` user ID. You cannot take any of the following actions:

- 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

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.

### About this task

To display command-line help for server commands that have unique names, you can type `help commandName`, where *commandName* is the name of the server command for which you want information. For example, to display help for the **REGISTER NODE** command, type `help register node`. Command syntax and parameter descriptions are displayed in the output.

You can also type `help` followed by the topic number for the command. Topic numbers are listed in the table of contents for command-line help, for example:

```
3.0 Administrative commands
  3.46 REGISTER
    3.46.1 REGISTER ADMIN (Register an administrator)
    3.46.2 REGISTER LICENSE (Register a new license)
    3.46.3 REGISTER NODE (Register a node)
```

To display help about the **REGISTER NODE** command, type:

```
help 3.46.3
```

Use topic numbers to display command-line help for subcommands. **DEFINE DEVCLASS** is an example of a command that has subcommands. For example, you can specify the **DEFINE DEVCLASS** command for 3590 device classes and for 3592 device classes:

3.0 Administrative commands

```
...
3.13.10 DEFINE DEVCLASS (Define a device class)
    3.13.10.1 DEFINE DEVCLASS (Define a 3590 device class)
    3.13.10.2 DEFINE DEVCLASS (Define a 3592 device class)
...
```

To display help for the **DEFINE DEVCLASS** command for 3590 device classes, type:  
help 3.13.10.1

## 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.

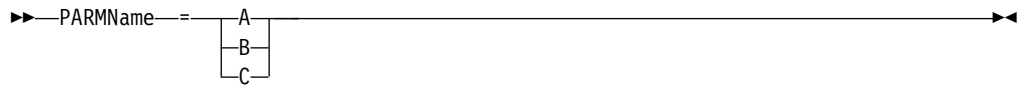
▶—CMDName—————▶◀

**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 **PARAMNAME=A**, **PARAMNAME=B**, or **PARAMNAME=C**. Do not include any blanks immediately before or after the equal sign (=).



## 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

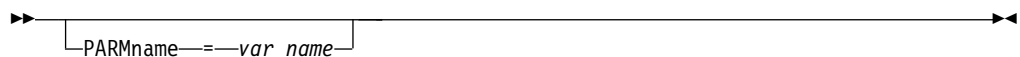
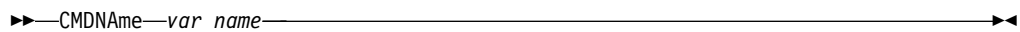
Defaults are above the line. The system uses the default unless you override it. You can override the default by entering an option from the stack below the line.

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 (=).



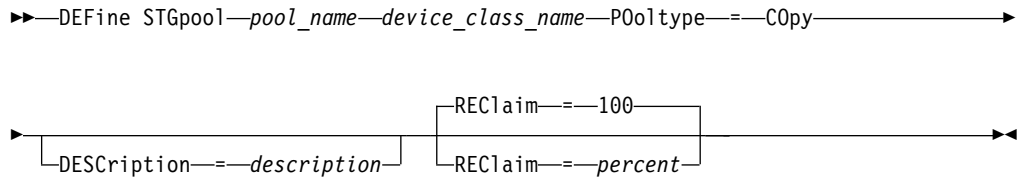
## Variables

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









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:

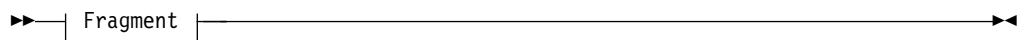
```
define stgpool mycopypool pooltype=copy mydeviceclass description=engineering
reclaim=50
```

### 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:



## Using continuation characters to enter 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.

### About this task

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 that are enclosed in quotation marks by entering the first part of the string that is 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

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

### About this task

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

The following characters are available for defining password names:

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
~ ! @ # $ % ^ & * _ - + = ` | ( ) { } [ ] : ; < > , . ? /
```

Passwords considered “LOCAL” are those password that authenticate with the Tivoli Storage Manager server and are not case-sensitive. Passwords considered “LDAP” are those password that authenticate with an LDAP directory server and are case-sensitive.

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 that 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.

### About this task

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

*Table 1. Wildcard characters by operating system*

Operating system	Match any	Match exactly one
AIX®, HP-UX, Linux, OS/2, Oracle 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 2 shows additional examples of using wildcard characters to match any characters.

*Table 2. 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 3 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 3. 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
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.

### About this task

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" <i>or</i> contact='manager's'
"manager"	contact=""manager"" <i>or</i> contact=""'manager'""
manager's report	contact="manager's report" <i>or</i> 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' <i>or</i> 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.

### About this task

## 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.

## Stopping background processes

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

### About this task

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.

### About this task

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.

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

### Procedure

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

### About this task

### Procedure

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

### About this task

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.

### Procedure

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

### About this task

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.

### Procedure

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

### About this task

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.

### Procedure

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

### About this task

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:

### Procedure

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
```

## Results

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 four 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 for the server. With system privilege, an administrator can issue any administrative command and has authority to manage all policy domains and all storage pools.

Table 4 on page 21 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 the server to write to an external file. For more information about this server option, review “REQSYSAUTHOUTFILE” on page 1547.

Table 4. System privilege commands

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

Table 4. System privilege commands (continued)

Command name	Command name
GRANT PROXYNODE	SET CONFIGMANAGER
IDENTIFY DUPLICATES	SET CONFIGREFRESH
IMPORT NODE	SET CONTEXTMESSAGING
IMPORT POLICY	SET CROSSDEFINE
IMPORT SERVER	SET DBRECOVERY
INSERT MACHINE	SET DEFAULTAUTHENTICATION
LABEL LIBVOLUME	SET DRMACTIVEDATASTGPOOL
LOCK ADMIN	SET DRMCHECKLABEL
LOCK PROFILE	SET DRMCMDFILENAME
MIGRATE STGPOOL	SET DRMCOPYSTGPOOL
MOVE DRMEDIA	SET DRMCOURIERNAME
MOVE MEDIA	SET DRMDBBACKUPEXPIREDAYS
MOVE GRPMEMBER	SET DRMFILPROCESS
NOTIFY SUBSCRIBERS	SET DRMINSTRPREFIX
PERFORM LIBACTION	SET DRMNOTMOUNTABLENAME
PING SERVER	SET DRMPPLANPREFIX
PREPARE	SET DRMPPLANVPOSTFIX
QUERY BACKUPSETCONTENTS	SET DRMPRIMSTGPOOL
QUERY MEDIA	SET DRMRPFEXPIREDAYS
QUERY RPFCONTENT	SET DRMVaultNAME
QUERY TOC	SET EVENTRETENTION
RECLAIM STGPOOL	SET INVALIDPWLIMIT
RECONCILE VOLUMES	SET LDAPPASSWORD
REGISTER ADMIN	SET LDAPUSER
REGISTER LICENSE	SET LICENSEAUDITPERIOD
REMOVE ADMIN	SET MAXCMDRETRIES
REMOVE REPLNODE	SET MAXSCHEDSESSIONS
RENAME ADMIN	SET MINPWLENGTH
RENAME SCRIPT	SET PASSEXP
RENAME SERVERGROUP	SET QUERYSCHEDPERIOD
RENAME STGPOOL	SET RANDOMIZE
REPLICATE NODE	SET REGISTRATION
RESET PASSEXP	SET REPLRETENTION
RESTORE NODE	SET REPLSERVER
REVOKE AUTHORITY	SET RETRYPERIOD
REVOKE PROXYNODE	SET SCHEDMODES
RUN	SET SERVERHLADDRESS
SET ACCOUNTING	SET SERVERLLADDRESS
SET ACTLOGRETENTION	SET SERVERNAME
SET ARCHIVERETENTIONPROTECTION	SET SERVERPASSWORD
SET AUTHENTICATION	SET SPREPLRULEDEFAULT
SET ARREPLRULEDEFAULT	SET SUBFILE
SET BKREPLRULEDEFAULT	SET TOCLOADRETENTION
SET CLIENTACTDURATION	

Table 4. System privilege commands (continued)

Command name	Command name
SETOPT	UPDATE NODEGROUP
UNLOCK ADMIN	UPDATE PATH
UNLOCK PROFILE	UPDATE PROFILE
UPDATE ADMIN	UPDATE RECOVERYMEDIA
UPDATE BACKUPSET	UPDATE REPLRULE
UPDATE CLIENTOPT	UPDATE SCHEDULE (Review note.)
UPDATE CLOPTSET	UPDATE SCRIPT
UPDATE COLLOCGROUP	UPDATE SERVER
UPDATE DEVCLASS	UPDATE SERVERGROUP
UPDATE DRIVE	UPDATE SPACETRIGGER
UPDATE LIBRARY	UPDATE VIRTUALFSMAPPING
UPDATE LIBVOLUME	UPDATE VOLHISTORY
UPDATE MACHINE	VALIDATE LANFREE
	VALIDATE REPLICATION

**Note:** This command is restricted by the authority that is 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.

With unrestricted policy privilege, you can 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.

With restricted policy privilege, you can issue administrator commands that affect one or more policy domains for which authority is granted. For example, the DELETE MGMTCLASS command requires you to have policy privilege for the policy domain to which the management class belongs.

Table 5 on page 24 lists the commands that an administrator with policy privilege can issue.

Table 5. Policy privilege commands

Command name	Command name
ACTIVATE POLICYSET	DELETE POLICYSET
ASSIGN DEFMGMTCLASS	DELETE PATH
CLEAN DRIVE	DELETE SCHEDULE (Review note 2.)
BACKUP NODE	GENERATE BACKUPSET
COPY MGMTCLASS	LOCK NODE
COPY POLICYSET	QUERY BACKUPSETCONTENTS
COPY SCHEDULE (Review 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 (Review note 1.)	UPDATE POLICYSET
DELETE FILESPACE	UPDATE SCHEDULE (Review 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 that is 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 6 lists the commands an administrator with storage privilege can issue.

*Table 6. Storage privilege commands*

Command name	Command name
AUDIT LIBRARY	DELETE SPACETRIGGER
AUDIT VOLUME (Review note.)	DELETE VIRTUALFSMAPPING
BACKUP DB	DELETE VOLHISTORY
BACKUP DEVCONFIG	DELETE VOLUME (Review note.)
BACKUP STGPOOL	GRANT PROXYNODE
BACKUP VOLHISTORY	LABEL LIBVOLUME
CHECKIN LIBVOLUME	MIGRATE STGPOOL
CHECKOUT LIBVOLUME	MOVE DATA (Review note.)
COPY ACTIVATEDATA (Review 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 (Review note.)	UPDATE DEVCLASS
DEFINE SPACETRIGGER	UPDATE DRIVE
DELETE COLLOGROUP	UPDATE LIBRARY
DELETE COLLOCMEMBER	UPDATE PATH
DELETE DATAMOVER	UPDATE SPACETRIGGER
DELETE DEVCLASS	UPDATE STGPOOL (Review 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 7 on page 26 lists the commands an administrator with operator privilege can issue.

*Table 7. 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 8 on page 27 lists the commands any registered administrator can issue.



Table 8. 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 REPLICATION
QUERY CONTENT	QUERY REPLNODE
QUERY COPYGROUP	QUERY REPLRULE
QUERY DATAMOVER	QUERY REQUEST
QUERY DB	QUERY RESTORE
QUERY DBSPACE	QUERY RPFIL
QUERY DEVCLASS	QUERY SCHEDULE
QUERY DIRSPACE	QUERY SCRIPT
QUERY DOMAIN	QUERY SERVER
QUERY DRIVE	QUERY SERVERGROUP
QUERY DRMEDIA	QUERY SESSION
QUERY DRMSTATUS	QUERY SPACETRIGGER
QUERY ENABLED	QUERY STATUS
QUERY EVENT	QUERY STGPOOL
QUERY EVENTRULES	QUERY SUBSCRIBER
QUERY EVENTSERVER	QUERY SUBSCRIPTION
QUERY FILESPACE	QUERY SYSTEM
QUERY LIBRARY	QUERY VIRTUALFSMAPPING
QUERY LIBVOLUME	QUERY VOLHISTORY
QUERY LICENSE	QUERY VOLUME
QUERY LOG	QUIT
QUERY MACHINE	ROLLBACK
QUERY MGMTCLASS	SELECT
QUERY MOUNT	SERIAL
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.

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 9. Command related to **ACCEPT DATE**

Command	Description
ENABLE SESSIONS	Resumes server activity following the <b>DISABLE</b> command or the <b>ACCEPT DATE</b> 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.

## 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 10. 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.

### 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 11. 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.
DEFINE POLICYSET	Defines a policy set within the specified policy domain.

Table 11. Commands related to ASSIGN DEFMGMTCLASS (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 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 **AUDIT LDAPDIRECTORY** command deletes nodes or administrator IDs from an LDAP directory server, that do not authenticate their passwords with the LDAP directory server.

- “AUDIT LDAPDIRECTORY (Audit an LDAP directory server)” on page 36
- “AUDIT LIBRARY (Audit volume inventories in an automated library)” on page 38
- “AUDIT LIBVOLUME (Verify database information for a tape volume)” on page 40
- “AUDIT LICENSES (Audit server storage usage)” on page 42
- “AUDIT VOLUME (Verify database information for a storage pool volume)” on page 43

## AUDIT LDAPDIRECTORY (Audit an LDAP directory server)

Use this command to audit an IBM Tivoli Storage Manager-controlled namespace on a Lightweight Directory Access Protocol (LDAP) server. The LDAP directory server and namespace are specified through one or more **LDAPURL** options.

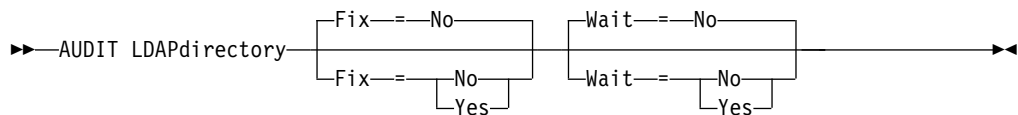
Nodes and administrator user IDs that do not authenticate their passwords with the LDAP directory server are deleted with the **AUDIT LDAPDIRECTORY FIX=YES** command. Nodes or administrator user IDs that no longer exist in the Tivoli Storage Manager database are also deleted.

Before you issue this command, ensure that the **LDAPURL** option is specified in the `dsmserv.opt` file. See the **LDAPURL** option for more information. If you specified more than one **LDAPURL** option in the `dsmserv.opt` file, each option is validated in the order in which they are placed. If the **LDAPURL** option is not specified, the command fails.

### Privilege class

You must have system privileges to issue this command.

### Syntax



### Parameters

#### Fix

This optional parameter specifies how the Tivoli Storage Manager server resolves inconsistencies between the database and the external directory. The default is **NO**. You can specify the following values:

**No** The server reports all inconsistencies but does not change the external directory.

#### Yes

The server resolves any inconsistencies that it can and suggests further actions, if needed.

**Important:** If there are LDAP entries that are shared with other Tivoli Storage Manager servers, choosing **YES** might cause those servers to become out-of-sync.

#### Wait

This optional parameter specifies whether to wait for the Tivoli Storage Manager server to complete processing this command in the foreground. The default is **NO**. You can specify the following values:

**No** The server processes this command in the background and you can continue with other tasks while the command is processing. Messages related to the background process are shown either in the activity log file or the server console, depending on where the messages are logged.

#### Yes

The server processes this command in the foreground. The operation must

complete before you can continue with other tasks. Messages are shown either in the activity log file or the server console, or both, depending on where the messages are logged.

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

### Example: Audit an LDAP directory and repair inconsistencies

Audit the LDAP directory that you specified in the LDAPURL option. The Tivoli Storage Manager server resolves some inconsistencies.

```
audit ldapdirectory fix=yes

ANR2749W Admin ADMIN1 was located in the LDAP directory server but not in the database.
ANR2749W Admin ADMIN2 was located in the LDAP directory server but not in the database.
ANR2749W Admin NODE1 was located in the LDAP directory server but not in the database.
ANR2749W Admin NODE2 was located in the LDAP directory server but not in the database.
ANR2748W Node NODE1 was located in the LDAP directory server but not in the database.
ANR2748W Node NODE2 was located in the LDAP directory server but not in the database.
ANR2745I AUDIT LDAPDIRECTORY command completed: 4 administrator entries are only in the
LDAP directory server (not in the Tivoli Storage Manager server), 0 administrator entries
are only in the Tivoli Storage Manager server (not in the LDAP directory server), 2 node
entries are only in the LDAP directory server (not in the Tivoli Storage Manager server),
0 node entries are only in the Tivoli Storage Manager server, (not in the LDAP directory
server), 6 entries were deleted from the LDAP server in total.
```

### Related commands

Table 12. Commands related to AUDIT LDAPDIRECTORY

Command	Description
SET DEFAULTAUTHENTICATION	Specifies the default password authentication method for any REGISTER NODE or REGISTER ADMIN commands.
SET LDAPPASSWORD	Sets the password for the LDAPUSER.
SET LDAPUSER	Sets the user who oversees the passwords and administrators on the LDAP directory server.

## 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.

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

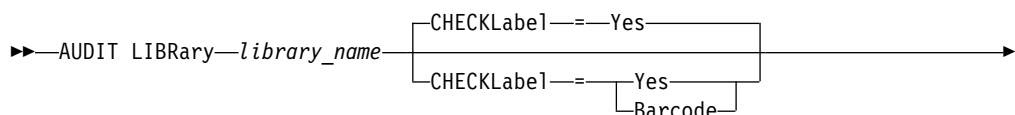
- Running the **AUDIT LIBRARY** command prevents any other library activity until the audit completes. For example, the server will not process restore or retrieve requests that involve the library when the **AUDIT LIBRARY** command is running.
- If other activity is occurring in the library, do not 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 the server checks each volume label to verify the identity of the volume.

#### Barcode

Specifies that the server 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, the server loads that tape in a drive to read the label.

### REFRESHstate

Specifies whether the server's information about a library, which is normally obtained during initialization, is refreshed, so that any changes in configuration are reflected. By setting the REFRESHSTATE parameter to Yes, this action is completed without having to restart the server or re-define the library. The default is No. Possible values are:

**No** Specifies that the server does not refresh the library's state when the library is audited.

#### Yes

Specifies that the server does refresh the library's state when the **AUDIT LIBRARY** command is issued.

## Example: Audit an automated library

Audit the EZLIFE automated library.

```
audit library ezlife
```

## Related commands

Table 13. 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.

Table 13. Commands related to **AUDIT LIBRARY** (continued)

Command	Description
QUERY LIBVOLUME	Displays information about a library volume.
QUERY PROCESS	Displays information about background processes.
UPDATE LIBRARY	Changes the attributes of a library.

## AUDIT LIBVOLUME (Verify database information for a tape volume)

Use this command to determine whether a tape volume is intact and to audit data on any tape volume.

You can issue the **AUDIT LIBVOLUME** command from any tape volume that is checked in to a library. The command runs in the background by default. You can issue the command from the following library types that have IBM TS1140, IBM LTO 5, or a later generation tape drive:

- SCSI tape library
- Virtual tape library (VTL)

The following table outlines the tape drives that can verify tape volumes with media types for IBM TS1140 and IBM LTO 5 and later generation LTO tape drives:

Table 14. Tape drives and the media types

Drive	Media type
TS1140	JB, JX, JA, JW, JJ, JR, JC, JY, and JK
IBM LTO 5	LTO 3, LTO 4, and LTO 5
IBM LTO 6	LTO 4, LTO 5, and LTO 6
IBM LTO 7	LTO 5, LTO 6, and LTO 7

The following table outlines the minimum device driver level that you require to run the command:

Table 15. Minimum IBM device driver level

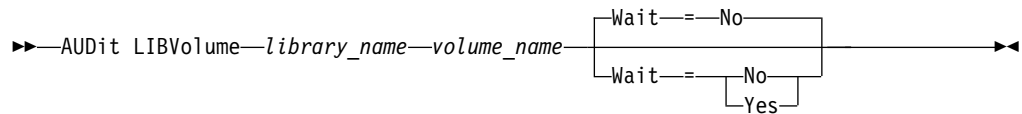
Driver name	Device driver level
Atape driver on AIX	12.3.5.00
lin_tape driver on Linux	1.6.7.00
IBM tape driver on Windows	6.2.2.00
IBM tape driver on Oracle Solaris SPARC	4.2.5.00
ATDD driver on HP 11i v3 IA64	6.0.2.17

**Restriction:** You cannot issue the **CANCEL PROCESS** command while the **AUDIT LIBVOLUME** command is in progress.

### Privilege class

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

## Syntax



## Parameters

### *library\_name* (Required)

Specifies the name of the library volume where the tape volume is located that you want to audit.

### *volume\_name* (Required)

Specifies the name of the physical tape volume that you want to audit.

### **Wait** (Optional)

Specifies whether the audit or verification operation is completed in the foreground or background. This parameter is optional. The following options are available:

**No** Specifies that the operation is completed in the background. The NO value is the default value.

#### **Yes**

Specifies that the operation is completed in the foreground. It might take a long time to complete the operation.

## Example: Audit a tape volume

Audit the EZLIFE library that has a tape volume that is called KM0347L5.

```
audit libvolume ezlife KM0347L5
```

## 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 16. 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 license with the 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.

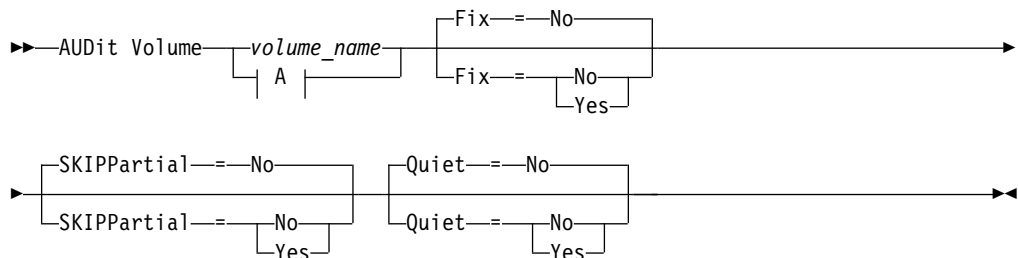
**Attention:** Use the `FIX=Yes` parameter only if your tape drive and storage area network (SAN) infrastructure is stable. Ensure that the tape heads are clean and that the tape device drivers are stable and reliable. Otherwise, you risk deleting data that is error free when you use this parameter. The Tivoli Storage Manager server cannot determine whether a tape is physically damaged or whether a tape infrastructure is unstable.

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.

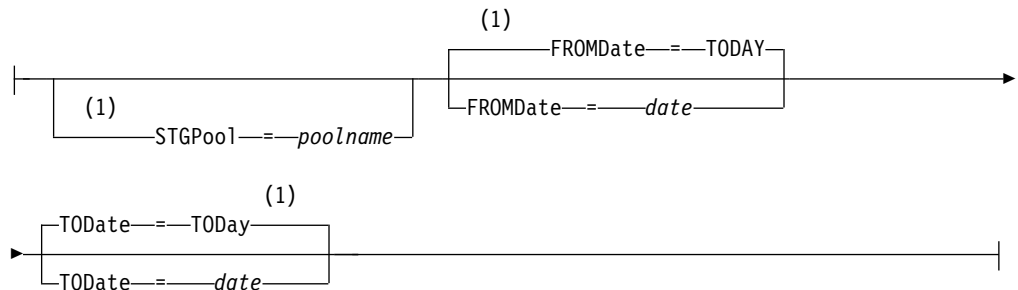
## Privilege class

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 partial 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 partial 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 partial 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:

<b>Value</b>	<b>Description</b>	<b>Example</b>
<i>MM/DD/YYYY</i>	A specific date	10/15/2001
<i>TODAY</i>	The current date	TODAY
<i>TODAY-days</i> or <i>-days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY -7 or -7.  To display information beginning with volumes written a week ago, you can specify FROMDATE=TODAY-7 or FROMDATE= -7.

Value	Description	Example
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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
MM/DD/YYYY	A specific date	10/15/2001  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- <i>days</i> or - <i>days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY-1 or -1.  To display information created up to yesterday, you can specify TODATE=TODAY-1 or simply TODATE= -1.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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 17. Commands related to AUDIT VOLUME*

<b>Command</b>	<b>Description</b>
CANCEL PROCESS	Cancels a background server process.
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** commands to create backup copies of Tivoli Storage Manager information or objects.

- “BACKUP DB (Back up the database)” on page 50
- “BACKUP DEVCONFIG (Create backup copies of device configuration information)” on page 55
- “BACKUP NODE (Back up a NAS node)” on page 57
- “BACKUP STGPOOL (Back up primary storage pool data to a copy storage pool)” on page 62
- “BACKUP VOLHISTORY (Save sequential volume history information)” on page 66

## BACKUP DB (Back up the database)

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

**Attention:** To restore a database, the server must use information from the volume history file and the device configuration file. You must make and save copies of the volume history file and the device configuration file. These files cannot be recreated.

To determine how much extra storage space a backup requires, issue the **QUERY DB** command.

**Restrictions:** You cannot restore a server database if the release level of the server database backup is different from the release level of the server that is being restored. For example, an error occurs when you restore a Version 6.3 database and you are using a Version 7.1 Tivoli Storage Manager server.

You cannot restore a server database from a different Tivoli Storage Manager server if both of the following conditions are present:

- The database backup is stored on virtual volumes
- The connection to the Tivoli Storage Manager server is protected by SSL (Secure Sockets Layer)

After the database backup is complete, the Tivoli Storage Manager server backs up information, depending on the options that are specified in the server options file. The following information is backed up:

- Sequential volume-history information is backed up to all files that the **VOLUMEHISTORY** option specifies
- Information about device configuration is backed up to all files that the **DEVCONFIG** option specifies

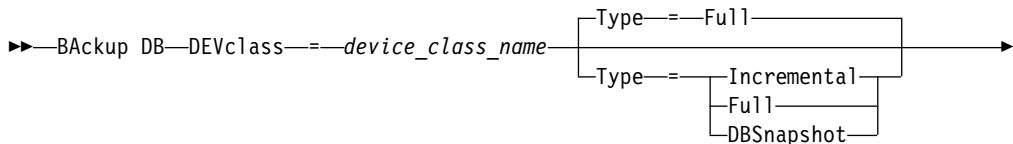
If there is not enough space available on the defined active log directory volume or file space, you can define the DB2<sup>®</sup> option, *overflowlogpath*, to use a directory with the required space available. For example, use the following command to use the `/home/tsminst2/overflow_dir` directory:

```
db2 update db cfg for TSMDB1 using overflowlogpath /home/tsminst2/overflow_dir
```

### Privilege class

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

### Syntax







### **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 that are 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 that are 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 that are 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
```

### **NUMStreams**

Specifies the number of parallel data movement streams to use when you back up the database. The minimum value is 1, and the maximum value is 4. Increasing the value causes a corresponding increase in the number of database backup sessions to be used and the number of drives to be used for the device class. If you specify a **NUMSTREAMS** value in the **BACKUP DB** command, it overrides any value that is set in the **SET DBRECOVERY** command. Otherwise, the value that is set in the **SET DBRECOVERY** command is used. The **NUMSTREAMS** value is used for all types of database backups.

If a value is specified that is greater than the number of drives available for the device class, only the number of available drives are used. The available drives are those defined to the device class by the **MOUNTLIMIT** parameter or by the number of online drives for the specified device class. The session is displayed in the **QUERY SESSION** output.

If you increase the number of streams, more volumes are used from the corresponding device class for this operation. Using more volumes might improve the speed of the database backups, but at the cost of more volumes that are not fully used.

### **Scratch**

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

#### **Yes**

Specifies that scratch volumes can be used.

If you specify **SCRATCH=YES** and the **VOLUMENAMES** parameter, Tivoli Storage Manager uses only 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. The following values are possible:

**No** Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

Messages that are 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 might 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 you continue 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.

#### **DEDUPDEVICE**

Specifies that a target storage device supports data deduplication. When set to YES, the format for backup images is optimized for data deduplication devices, making backup operations more efficient. The following values are possible:

**No** Specifies that a target storage device does not support data deduplication. NO is the default.

Ensure that this parameter is set to NO for the following devices:

- SCSI libraries
- All devices that are defined with a FILE device class
- Virtual tape libraries (VTL) that do not support the data deduplication function

#### **Yes**

Specifies that a target device supports data deduplication and that you want to optimize backups for this function. You can set this parameter to YES if you are using VTLs that support data deduplication.

#### **COMPRESS**

Specifies whether volumes that are created by the **BACKUP DB** command are compressed. The **COMPRESS** value is used for all types of database backups. This parameter is optional. The default value is conditional. If you specify the **COMPRESS** parameter on the **BACKUP DB** command, it overrides any value that is

set in the **SET DBRECOVERY** command. Otherwise, the value that is set in the **SET DBRECOVERY** command is the default. You can specify one of the following values:

**No** Specifies that the volumes that are created by the **BACKUP DB** command are not compressed.

**Yes**

Specifies that the volumes that are created by the **BACKUP DB** command are compressed.

**Restrictions:**

- Use caution when you specify the **COMPRESS** parameter. Using compression during database backups can reduce the size of the backup files. However, compression can increase the time that is required to complete database backup processing.
- Do not back up compressed data to tape. If your system environment stores database backups on tape, set the **COMPRESS** parameter to No in the **SET DBRECOVERY** and **BACKUP DB** commands.

**Example: Run an incremental backup by using a scratch volume**

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

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

**Related commands**

*Table 18. Commands related to **BACKUP DB***

Command	Description
BACKUP DEVCONFIG	Backs up 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 onsite and offsite.
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 DBRECOVERY	Specifies the device class to be used for automatic backups.
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.

**Attention:** To restore a database, the server must use information from the volume history file and the device configuration file. You must make and save copies of the volume history file and the device configuration file. These files cannot be recreated.

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.

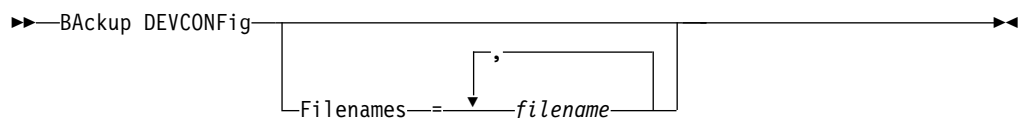
To ensure updates are complete before the server is halted:

- 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 19. 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)

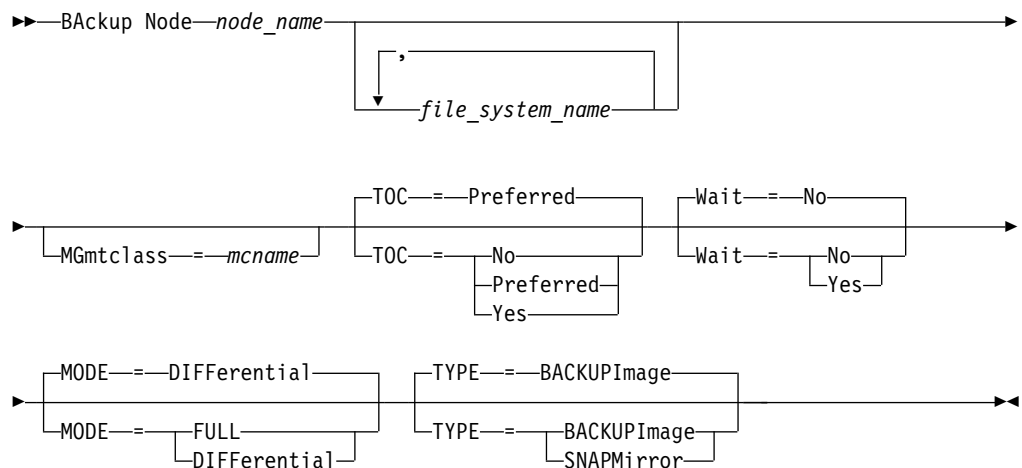
Use this command to start a backup operation for a network-attached storage (NAS) node.

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.

When you specify a management class with the BACKUP NODE command, all versions of the backup data that belong to the NAS node are rebound to the new management class.

#### **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.

For most NetApp file systems, use the standard NDMP full or differential 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 20 contains the virtual file space definitions exist on the server for the node NAS1.

```
backup node nas1 /mikesdir
```

Table 20. 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 20 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 21. Commands related to BACKUP NODE

Command	Description
BACKUP NAS (client command)	Creates a backup of NAS node data.
CANCEL PROCESS	Cancels a background server process.
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 21. 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 (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 data to a copy storage pool)

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

You can back up data from a primary storage pool that is defined with the NATIVE, NONBLOCK, or any of the NDMP 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 back-end data movement for NDMP images.

If a file 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, cached copies of migrated files and damaged primary files are not backed up.

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

If migration for a storage pool starts during a storage pool backup, some files might be migrated before they are backed up. You might want to back up storage pools that are higher in the migration hierarchy before you back up storage pools that are lower.

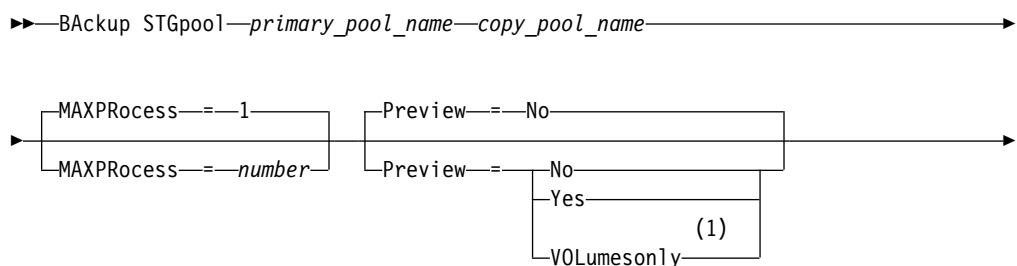
### Restrictions:

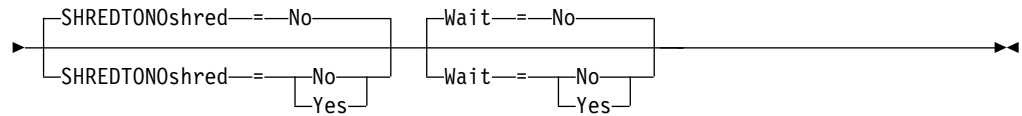
- Do not run the **MOVE DRMEDIA** and **BACKUP STGPOOL** commands concurrently. Ensure that the storage pool backup processes are complete before you issue the **MOVE DRMEDIA** command.
- You cannot back up data from or to storage pools defined with a CENTERA device class.

### 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 that are 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 1 - 999. The default is 1.

Using multiple, parallel processes can improve throughput for the backup. The expectation is that the time needed to complete the storage pool backup is 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 you determine 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 extra mount point for primary storage pool volumes and, if the device type is not FILE, an extra drive. For example, suppose that 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 six mount points and six drives must be available.

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

**Preview**

Specifies whether you want to preview but not run 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. You can specify the following values:

**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 that are associated with a sequential-access device class.

The VOLUMESONLY option can be used to obtain a list of volumes that are needed by the storage pool backup process. For example:

```
backup stgpool primary_pool copystg preview=volumesonly
```

The list of volumes are logged in the server activity log with the ANR1228I message. Query the server activity log to get the list of volumes required. For example:

```
query actlog msg=1228
```

#### **SHREDTONOshred**

Specifies whether data is backed up to a copy storage pool from a primary storage pool that enforces shredding. This parameter is optional. The default value is NO. You can specify the following values:

**No** Specifies that the server does 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 fails.

#### **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 is not 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. You can specify the following values:

**No** Specifies that the server processes this command in the background.

You can continue with other tasks while the command is being processed. Messages that are 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 might already have been backed up before the cancellation.

#### **Yes**

Specifies that the server processes this operation in the foreground. You must wait for the operation to complete before you continue 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: Back up the primary storage pool**

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

```
backup stgpool primary_pool copystg
```

## Related commands

Table 22. Commands related to **BACKUP STGPOOL**

Command	Description
CANCEL PROCESS	Cancels a background server process.
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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.
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

**Attention:** To restore a database, the server must use information from the volume history file and the device configuration file. You must make and save copies of the volume history file and the device configuration file. These files cannot be recreated.

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:

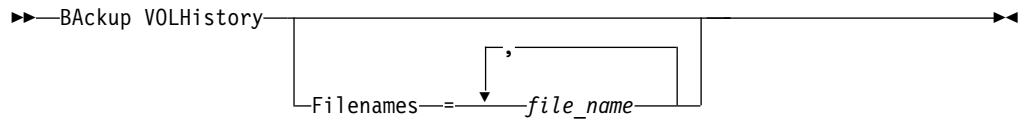
- Not halt the server for a few minutes after issuing the BACKUP VOLHISTORY command.
- Specify multiple VOLUMEHISTORY options in the server options file.
- Examine the volume history file to see if the file has been updated.

### 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 23. 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)

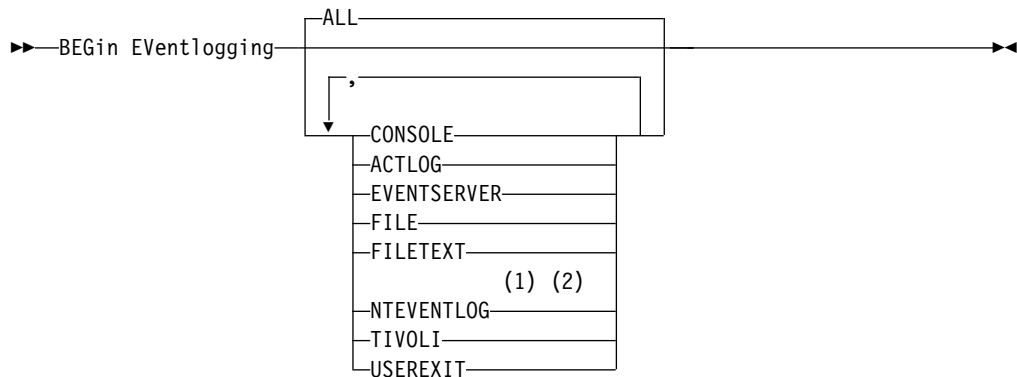
Use this command to begin logging events to one or more receivers. A receiver for which event logging has begun is an *active receiver*.

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 the Windows operating system.
- 2 This parameter is only available for the Linux operating system.

### 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 24. Commands related to BEGIN EVENTLOGGING*

<b>Command</b>	<b>Description</b>
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 or process before it is completed.

- “CANCEL EXPIRATION (Cancel an expiration process)” on page 71
- “CANCEL EXPORT (Delete a suspended export operation)” on page 72
- “CANCEL PROCESS (Cancel an administrative process)” on page 73
- “CANCEL REPLICATION (Cancel node replication processes)” on page 75
- “CANCEL REQUEST (Cancel one or more mount requests)” on page 76
- “CANCEL RESTORE (Cancel a restartable restore session)” on page 77
- “CANCEL SESSION (Cancel one or more client sessions)” on page 78

## CANCEL EXPIRATION (Cancel an expiration process)

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

Use the CANCEL EXPIRATION command if the expiration process number is not known, otherwise use the CANCEL PROCESS and specify the process number of the expiration process. Both commands call the same code to end the expiration process.

You can use the CANCEL EXPIRATION command to automate the cancellation of an expiration process. For example, if you start inventory expiration at midnight and, due to the maintenance workload on the server, the process must finish at 03:00, you can schedule a CANCEL EXPIRATION command to run at 03:00 without knowing the process number.

### 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 25. Command related to CANCEL EXPIRATION

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

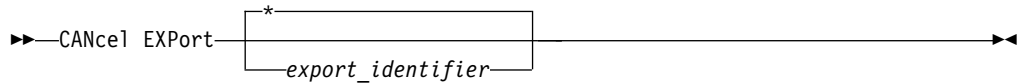
## CANCEL EXPORT (Delete a suspended export operation)

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 26. Commands related to **CANCEL EXPORT**

Command	Description
CANCEL PROCESS	Cancels a background server process.
EXPORT NODE	Copies client node information to external media or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
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
- PROTECT STGPOOL
- RECLAIM STGPOOL
- REPLICATE NODE
- 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, generate mount requests to complete processing. If a process has a pending mount request, the process might not respond to a **CANCEL PROCESS** command until the mount request is answered or canceled by using 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 whether a 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. The volume might not be available if the administrator issues the **MOVE MEDIA** or **CHECKOUT LIBVOLUME** command, or manually removes 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 later, issue the **SUSPEND EXPORT** command.

## Privilege class

To issue this command, you must have system privilege.

## 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 by using its process number

Cancel background process number 3.

```
cancel process 3
```

## Related commands

Table 27. 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.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
REPLY	Allows a request to continue processing.
RESTART EXPORT	Restarts a suspended export operation.
SUSPEND EXPORT	Suspends a running export operation.



## CANCEL REPLICATION (Cancel node replication processes)

Use this command to cancel all node replication processes.

Issue this command on the server that acts as a source for replicated data.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—CANCEL REPLICATION—◄◄

### Parameters

None.

### Example: Cancel node replication processes

Cancel all node replication processes.

```
cancel replication
```

### Related commands

*Table 28. Commands related to CANCEL REPLICATION*

Command	Description
QUERY PROCESS	Displays information about background processes.
QUERY REPLICATION	Displays information about node replication processes.

## 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

```
►►—cANcel REQuest—request_number—ALl—PERManent—►►
```

### 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 29. 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 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

```
►►—CANce1—REStore—session_number—►►  
                          └─AL1—┘
```

### 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.

**AL1**

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 30. 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] [AL1] ▶▶
```

### Parameters

*session\_number*

Specifies the number of the administrative, server, or client node sessions that you want to cancel.

**AL1**

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

### Example: Cancel a specific client node session

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

```
cancel session 3
```

### Related commands

Table 31. Commands related to CANCEL SESSION

Command	Description
DISABLE SESSIONS	Prevents new sessions from accessing Tivoli Storage Manager but permits existing sessions to continue.
LOCK ADMIN	Prevents an administrator from accessing Tivoli Storage Manager.
LOCK NODE	Prevents a client from accessing the server.
QUERY SESSION	Displays information about all active administrator and client sessions with 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 is 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.
3. When you check in WORM tapes other than 3592, you must use **CHECKLABEL=YES** or they are 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 about background processes, use the **QUERY PROCESS** command.

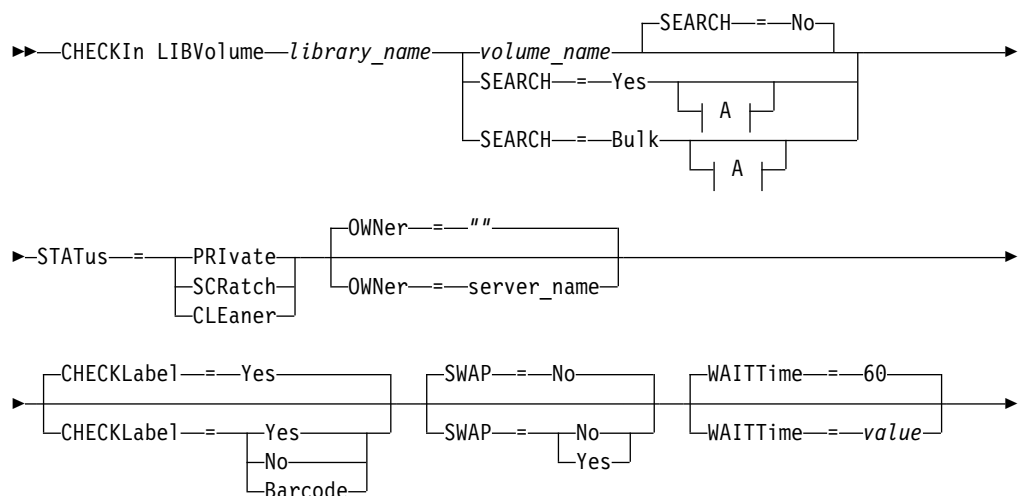
For detailed and current drive and library support information, see the Supported Devices website for your operating system:

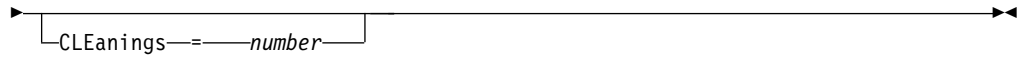
[http://www.ibm.com/software/sysmgmt/products/support/IBM\\_TSM\\_Supported\\_Devices\\_for\\_AIXHPSUNWIN.html](http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html)

### Privilege class

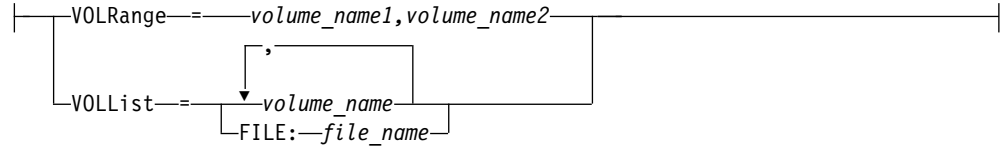
To issue this command, you must have system privilege or unrestricted storage privilege.

### Syntax for SCSI libraries

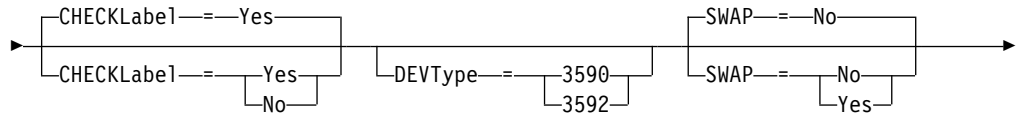
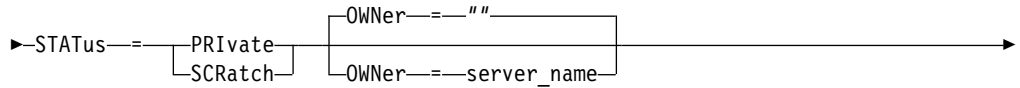
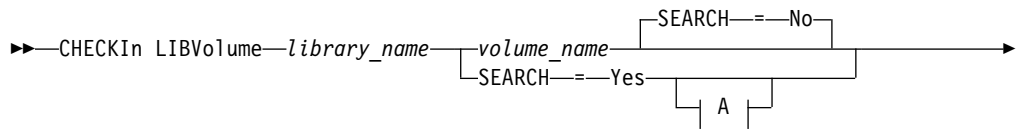




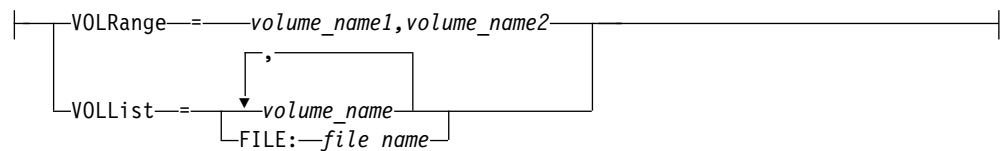
**A (SEARCH=Yes, SEARCH=Bulk):**



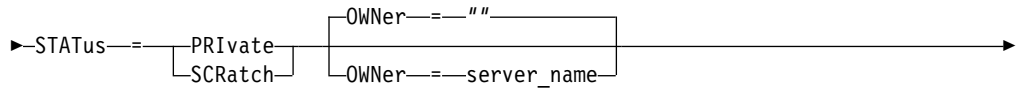
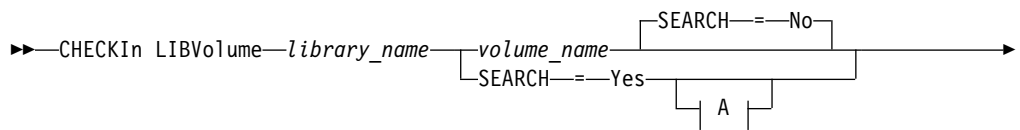
**Syntax for 349X libraries**

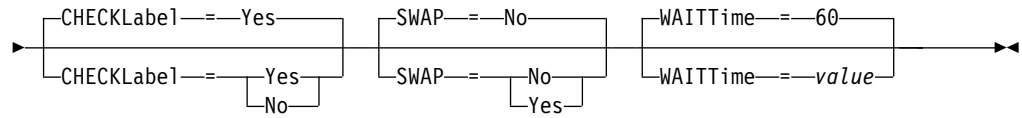


**A (SEARCH=Yes):**

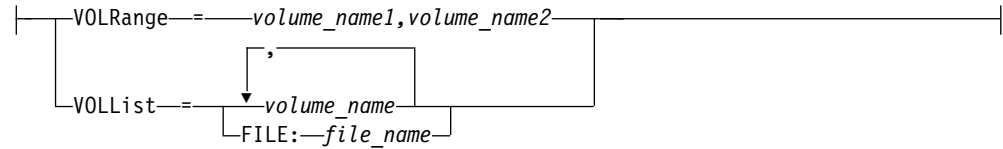


**Syntax for ACSLS libraries**





**A (SEARCH=Yes):**



**Parameters**

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

Specifies the name of the library.

*volume\_name*

Specifies the volume name of the storage volume that is 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 is 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 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 might already be in the library, or you can 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 you use this parameter, consider the following restrictions:

- If the library is shared between applications, the server might examine a volume that is 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 applies to only 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 that are 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:

*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 SCSI libraries, do not specify both SEARCH=YES and CHECKLABEL=NO in the same command.
2. 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 about 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 that is being checked in. This parameter is required if none of the drives in this library have defined paths.

**3590**

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

**3592**

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

**SWAP**

Specifies whether the server swaps volumes if an empty library slot is not available. The volume that is selected for the swap operation (target swap volume) is ejected from the library and replaced with the volume that is 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 applies only if there is a volume name that is 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 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 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
```

### 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 32. 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.

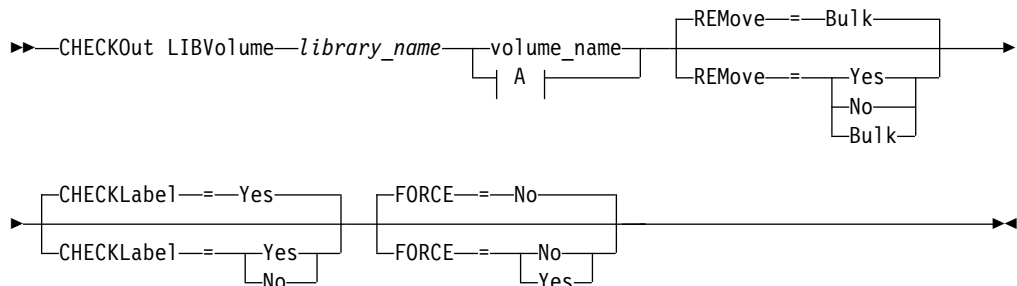
For detailed and current drive and library support information, see the Supported Devices Web site for your operating system:

[http://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)

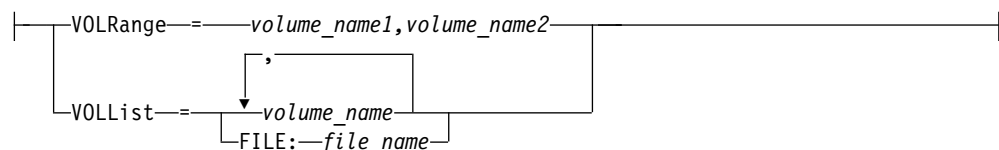
### Privilege class

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

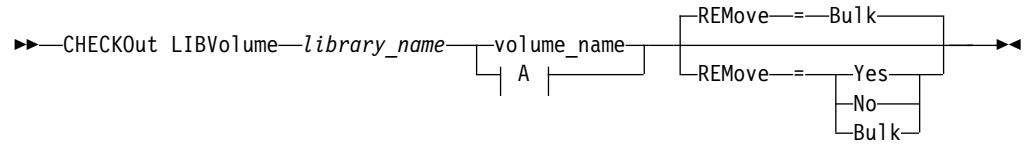
### Syntax for SCSI library



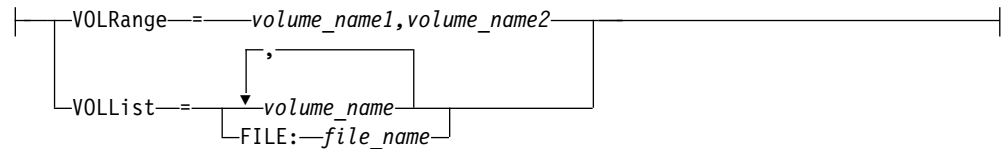
**A :**



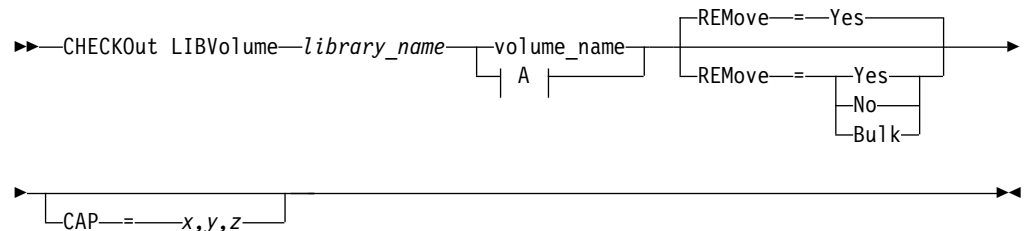
## Syntax for 349X library



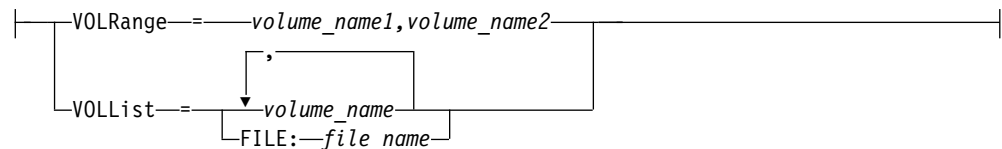
**A :**



## 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, 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 33. How the 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.  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.

Table 34. How the server responds for SCSI libraries

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

**ACSLs libraries:** The default is YES. If the parameter is set to YES, and the cartridge access port (CAP) has an automatic selection priority value of 0, you must specify a CAP ID. The following table shows how the server responds for ACSLS libraries.

Table 35. How the 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 the CAP priority value is set to 0 in the library, this parameter is required. If a CAP priority value greater than 0 is set in the library, this parameter is optional. By default, all CAPs initially have a priority value of 0, which means that ACSLS does not automatically select the CAP.

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 in the range 0 - 126.
- $y$  The Library Storage Module (LSM) ID. This identifier can be a number in the range 0 - 23.
- $z$  The CAP ID. This identifier can be a number in the range 0 - 11.

For more information, see the 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 36. 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.



Table 36. Commands related to CHECKOUT LIBVOLUME (continued)

<b>Command</b>	<b>Description</b>
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.

**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 37. 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 38. 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 or data.

- “COPY ACTIVE DATA (Copy active backup data from a primary storage pool to an active-data pool)” on page 95
- “COPY CLOPTSET (Copy a client option set)” on page 99
- “COPY DOMAIN (Copy a policy domain)” on page 100
- “COPY MGMTCLASS (Copy a management class)” on page 102
- “COPY POLICYSET (Copy a policy set)” on page 104
- “COPY PROFILE (Copy a profile)” on page 106
- “COPY SCHEDULE (Copy a client or an administrative command schedule)” on page 108
- “COPY SCRIPT (Copy a Tivoli Storage Manager script)” on page 112
- “COPY SERVERGROUP (Copy a server group)” on page 113

## 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.

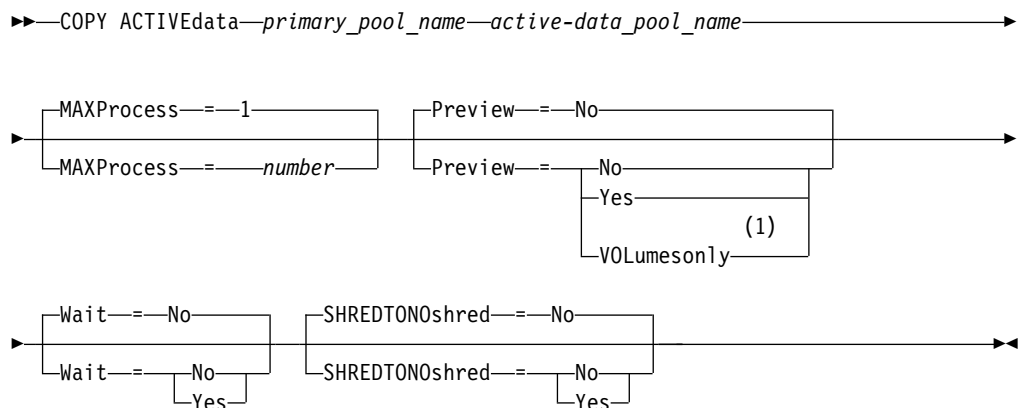
### Remember:

- 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 data deduplication removes duplicate data, if the active-data pool is also set up for data 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 ACTIVE** 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 ACTIVE** 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, the server 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 server and system activity, and also 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:

**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 39. Commands related to COPY ACTIVEDATA*

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 or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
IMPORT NODE	Restores client node information from external media.
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.

Table 39. Commands related to COPY ACTIVE DATA (continued)

Command	Description
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 40. 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 is 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 STANDARD policy domain to a new policy domain, ENGPOLDOM, by entering the following command:

```
copy domain standard engpoldom
```

ENGPOLDOM now contains the standard policy set, management class, backup copy group, and archive copy group.

### Related commands

Table 41. 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.

*Table 41. Commands related to COPY DOMAIN (continued)*

<b>Command</b>	<b>Description</b>
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.
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 node 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 42. 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.

Table 42. Commands related to **COPY MGMTCLASS** (continued)

<b>Command</b>	<b>Description</b>
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

The policies in the new policy set do not take effect unless you make the new set 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 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 43. 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.

Table 43. Commands related to **COPY POLICYSET** (continued)

Command	Description
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 44. 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.



Table 44. Commands related to **COPY PROFILE** (continued)

<b>Command</b>	<b>Description</b>
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.

- “COPY SCHEDULE (Create a copy of a schedule for client operations)” on page 109
- “COPY SCHEDULE (Create a copy of a schedule for administrative operations)” on page 111

*Table 45. Commands related to COPY SCHEDULE*

<b>Command</b>	<b>Description</b>
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)

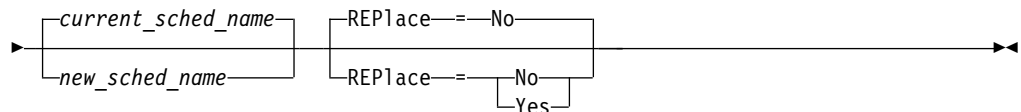
Use the **COPY SCHEDULE** command to create a copy of a schedule for client operations. You can copy a schedule within a policy domain or from one policy domain to another policy domain. Use the **DEFINE ASSOCIATION** command to associate the new schedule with the client nodes.

### 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 an existing Tivoli Storage Manager script to a new script with a different name.

### 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 46. Commands related to COPY SCRIPT

Command	Description
DEFINE SCRIPT	Defines a script to the 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 47. 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.

## DEACTIVATE DATA (Deactivate data for a client node)

Use this command to specify that active data that was backed up for an application client node before a specified date is no longer needed. The command marks the data as inactive so it can be deleted according to your data retention policies.

**Restriction:** The **DEACTIVATE DATA** command applies only to application clients that protect Oracle databases.

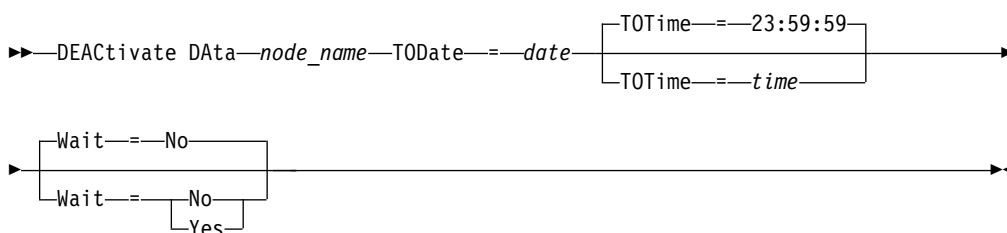
When you issue the **DEACTIVATE DATA** command, all active backup data that was stored before the specified date becomes inactive. The data can no longer be retrieved, and is deleted when it expires.

The **DEACTIVATE DATA** command affects only the files that were copied to the server before the specified date and time. Files that were copied after the specified date are still accessible, and the client can still access the server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

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

Specifies the name of an application client node whose data is to be deactivated.

#### *TDate* (Required)

Specifies the date to use to select the backup files to deactivate. Tivoli Storage Manager deactivates only those files with a date on or before 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	01/23/2014
TODAY	The current date	TODAY
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY-30 or -30. To deactivate files that are 30 or more days old, you can specify TODAY-30 or -30.
EOLM	End of last month. The last day of the previous month.	EOLM



Value	Description	Example
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To deactivate files that were active a day before the last day of the previous month.
BOTM	Beginning of this month. The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To deactivate files that were active on the 10th day of the current month.

### TOTime

Specifies that you want to deactivate files that were created on the server before this time on the specified date. This parameter is optional. The default is the end of the day (23:59:59). Specify the time by using one of the following values:

Value	Description	Example
<i>HH:MM:SS</i>	A specific time on the specified date	12:30:22
NOW	The current time on the specified date	NOW
<i>NOW+HH:MM or +HH:MM</i>	The current time plus hours and minutes on the specified date	NOW+03:00 or +03:00.  If you issue the <b>DEACTIVATE DATA</b> command at 9:00 with TOTIME=NOW+03:00 or TOTIME=+03:00, Tivoli Storage Manager deactivates files that were put on the server at 12:00 or earlier on the specified date.
<i>NOW-HH:MM or -HH:MM</i>	The current time minus hours and minutes on the specified date	NOW-03:30 or -03:30.  If you issue the <b>DEACTIVATE DATA</b> command at 9:00 with TOTIME=NOW-3:30 or TOTIME=-3:30, Tivoli Storage Manager deactivates files that were put on the server at 5:30 or earlier on the specified date.

### 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. Specify the following values:

- No** The server processes this command in the background and you can continue with other tasks while the command is processing. Messages that are related to the background process are shown either in the activity log file or the server console, depending on where the messages are logged.
- Yes** The server processes this command in the foreground. The operation must complete before you can continue with other tasks. Messages are

shown either in the activity log file or the server console, or both, depending on where the messages are logged.

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

### **Example: Deactivate data for a data protection client node**

The client node BANDIT is a Tivoli Storage Manager for Databases: Data Protection for Oracle application client. All of the backup data is active, and so all of the backup data is retained. The following command deactivates data that was backed up before January 3, 2014, so it can be deleted when it expires.

```
deactivate data bandit todate=01/23/2014
```

To periodically deactivate data so it can be deleted when it expires, you might run the following command from within a client schedule.

```
deactivate data bandit todate=today
```

### **Related commands**

*Table 48. Commands related to DEACTIVATE DATA*

<b>Command</b>	<b>Description</b>
DECOMMISSION NODE	Decommissions an application or system.
DECOMMISSION VM	Decommissions a virtual machine.

---

## DECOMMISSION commands

Use the **DECOMMISSION** commands to remove client nodes from the production environment. Client nodes include applications, systems, and virtual machines.

- “DECOMMISSION NODE (Decommission an application or system)” on page 118
- “DECOMMISSION VM (Decommission a virtual machine)” on page 121

## DECOMMISSION NODE (Decommission an application or system)

Use this command to remove an application or system client node from the production environment. Any backup data that is stored for the client node expires according to policy settings unless you explicitly delete the data.

**Attention:** This action cannot be reversed and causes deletion of data. Although this command does not delete the client node definition until after its data expires, you cannot recommission the client node. After you issue this command, the client node cannot access the server and its data is not backed up. The client node is locked, and can be unlocked only to restore files. File spaces that belong to the client node, and the client node itself, are eventually removed.

By using this command, you can decommission the following types of client nodes:

### Application client nodes

Application client nodes include email servers, databases, and other applications. For example, any of the following applications can be an application client node:

- IBM Tivoli Storage FlashCopy Manager
- IBM Tivoli Storage Manager for Databases
- IBM Tivoli Storage Manager for Enterprise Resource Planning
- IBM Tivoli Storage Manager for Mail
- IBM Tivoli Storage Manager for Virtual Environments

### System client nodes

System client nodes include workstations, network-attached storage (NAS) file servers, and API clients.

When a client node is no longer needed in the production environment, you can issue this command to initiate a gradual, controlled decommission operation. The command completes the following actions:

- Deletes all schedule associations for the client node. Schedules are no longer run on the client node. This action is equivalent to issuing the **DELETE ASSOCIATION** command for every schedule with which the client node is associated.
- Prevents the client from accessing the server. This action is equivalent to issuing the **LOCK NODE** command.

After the command finishes, client node data is no longer backed up to the server. Data that was backed up before the client node was decommissioned is not immediately deleted from the server. However, all backup file versions, including the most recent backup, are now inactive copies. The client files are retained on the server according to your storage management policies.

After all data retention periods expire, and all client backup and archive file copies are removed from server storage, Tivoli Storage Manager deletes the file spaces that belong to the decommissioned node. This action is equivalent to issuing the **DELETE FILESPACE** command.

After the file spaces for the decommissioned node are deleted, the node definition is deleted from the server. This action is equivalent to issuing the **REMOVE NODE** command.

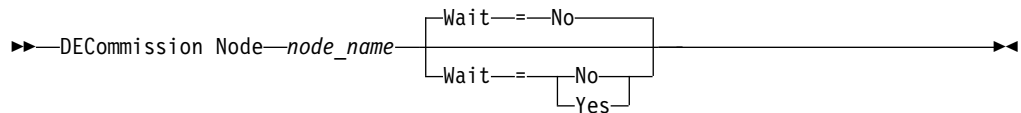
After you decommission a client node, but before it is removed from the server, you can use the **QUERY NODE** command to verify that the client node is decommissioned.

**Restriction:** You cannot decommission a client node that is configured for replication. You can determine a client node's replication state by using the **QUERY NODE** command. If a client node is configured for replication, you can remove the client node from replication by using the **REMOVE REPLNODE** 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 client node belongs.

## Syntax



## Parameters

### *node\_name* (Required)

Specifies the name of the client node to be decommissioned.

### 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. You can specify the following values:

- No** The server processes this command in the background and you can continue with other tasks while the command is processing. Messages that are related to the background process are shown either in the activity log file or the server console, depending on where the messages are logged.
- Yes** The server processes this command in the foreground. The operation must complete before you can continue with other tasks. Messages are shown either in the activity log file or the server console, or both, depending on where the messages are logged.

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

## Example: Decommission a client node

Decommission the client node CODY.

```
decommission node cody
```

## Related commands

Table 49. Commands related to **DECOMMISSION NODE**

Command	Description
DECOMMISSION VM	Decommissions a virtual machine.
DEACTIVATE DATA	Deactivates data for a client node.



## DECOMMISSION VM (Decommission a virtual machine)

Use this command to remove an individual virtual machine within a data center node. The file space that represents the virtual machine is deleted from the server only after its backup data expires.

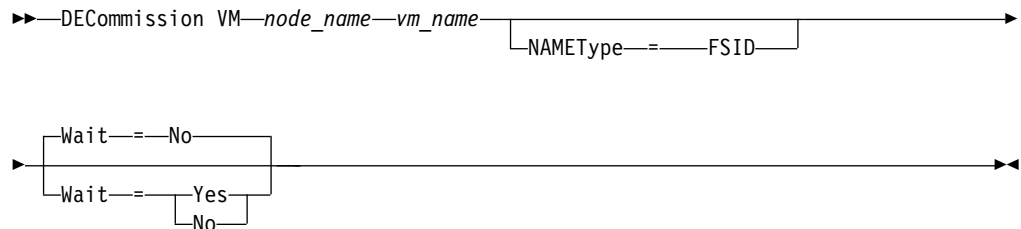
**Attention:** This command cannot be reversed and causes deletion of data. Although this command does not delete the virtual machine file space until after its data expires, you cannot recommission the virtual machine.

When a virtual machine is no longer needed in your production environment, you can issue this command to initiate a staged removal of the virtual machine file space from the server. The **DECOMMISSION VM** command marks all data that was backed up for the virtual machine as inactive, so it can be deleted according to your data retention policies. After all data that was backed up for the virtual machine expires, the file space that represents the virtual machine is deleted. The **DECOMMISSION VM** command affects only the virtual machine that you identify. The data center node, and the other virtual machines that are hosted by the data center node are not affected.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

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

Specifies the name of the data center node that hosts the virtual machine to be decommissioned.

#### *vm\_name* (Required)

Identifies the file space that represents the virtual machine to be decommissioned. Each virtual machine that is hosted by a data center node is represented as a file space.

If the name includes one or more spaces, you must enclose the name in double quotation marks when you issue the command.

By default, the server interprets the file space name that you enter by using the server code page and also attempts to convert the file space name from the server code page to the UTF-8 code page. Conversion might fail if the string includes characters that are not available in the server code page, or if the server cannot access system conversion routines.

If the name of the virtual machine is a non-English-language name, this parameter must specify the file space ID (FSID). By specifying the **NAMETYPE** parameter, you can instruct the server to interpret the file space name by its file space ID (FSID) instead.

**NAMETYPE**

Specify how you want the server to interpret the file space name that you enter to identify the virtual machine. This parameter is useful when the server has clients with Unicode support. You can specify the following value:

**FSID**

The server interprets the file space name by its file space ID (FSID).

**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. You can specify the following values:

- No** The server processes this command in the background and you can continue with other tasks while the command is processing. Messages that are related to the background process are shown either in the activity log file or the server console, depending on where the messages are logged.
- Yes** The server processes this command in the foreground. The operation must complete before you can continue with other tasks. Messages are shown either in the activity log file or the server console, or both, depending on where the messages are logged.

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

**Examples: Decommission a virtual machine**

Decommission the virtual machine CODY.

```
decommission vm dept06node cody
```

Decommission the virtual machine CODY 2.

```
decommission vm dept06node "cody 2"
```

Decommission a virtual machine by specifying its file space ID.

```
decommission vm dept06node 7 nametype=fsid
```

**Related commands**

*Table 50. Commands related to DECOMMISSION VM*

Command	Description
DECOMMISSION NODE	Decommissions an application or system.
DEACTIVATE DATA	Deactivates data for a client node.



---

## DEFINE commands

Use the **DEFINE** commands to create Tivoli Storage Manager objects.

- “DEFINE ALERTTRIGGER (Define an alert trigger)” on page 124
- “DEFINE ASSOCIATION (Associate client nodes with a schedule)” on page 127
- “DEFINE BACKUPSET (Define a backup set)” on page 129
- “DEFINE CLIENTACTION (Define a one-time client action)” on page 133
- “DEFINE CLIENTOPT (Define an option to an option set)” on page 139
- “DEFINE CLOPTSET (Define a client option set name)” on page 141
- “DEFINE COLLOGROUP (Define a collocation group)” on page 142
- “DEFINE COLLOCMEMBER (Define collocation group member)” on page 144
- “DEFINE COPYGROUP (Define a copy group)” on page 148
- “DEFINE DATAMOVER (Define a data mover)” on page 158
- “DEFINE DEVCLASS (Define a device class)” on page 161
- “DEFINE DOMAIN (Define a new policy domain)” on page 223
- “DEFINE DRIVE (Define a drive to a library)” on page 225
- “DEFINE EVENTSERVER (Define a server as the event server)” on page 230
- “DEFINE GRPMEMBER (Add a server to a server group)” on page 231
- “DEFINE LIBRARY (Define a library)” on page 232
- “DEFINE MACHINE (Define machine information for disaster recovery)” on page 253
- “DEFINE MACHNODEASSOCIATION (Associate a node with a machine)” on page 255
- “DEFINE MGMTCLASS (Define a management class)” on page 257
- “DEFINE NODEGROUP (Define a node group)” on page 260
- “DEFINE NODEGROUPMEMBER (Define node group member)” on page 261
- “DEFINE PATH (Define a path)” on page 262
- “DEFINE POLICYSET (Define a policy set)” on page 271
- “DEFINE PROFASSOCIATION (Define a profile association)” on page 273
- “DEFINE PROFILE (Define a profile)” on page 279
- “DEFINE RECMEDMACHASSOCIATION (Associate recovery media with a machine)” on page 281
- “DEFINE RECOVERYMEDIA (Define recovery media)” on page 283
- “DEFINE SCHEDULE (Define a client or an administrative command schedule)” on page 285
- “DEFINE SCRIPT (Define a Tivoli Storage Manager script)” on page 312
- “DEFINE SERVER (Define a server for server-to-server communications)” on page 315
- “DEFINE SERVERGROUP (Define a server group)” on page 322
- “DEFINE SPACETRIGGER (Define the space trigger)” on page 323
- “DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)” on page 326
- “DEFINE STGPOOL (Define a storage pool)” on page 330
- “DEFINE SUBSCRIPTION (Define a profile subscription)” on page 377
- “DEFINE VIRTUALFSMAPPING (Define a virtual file space mapping)” on page 379
- “DEFINE VOLUME (Define a volume in a storage pool)” on page 381

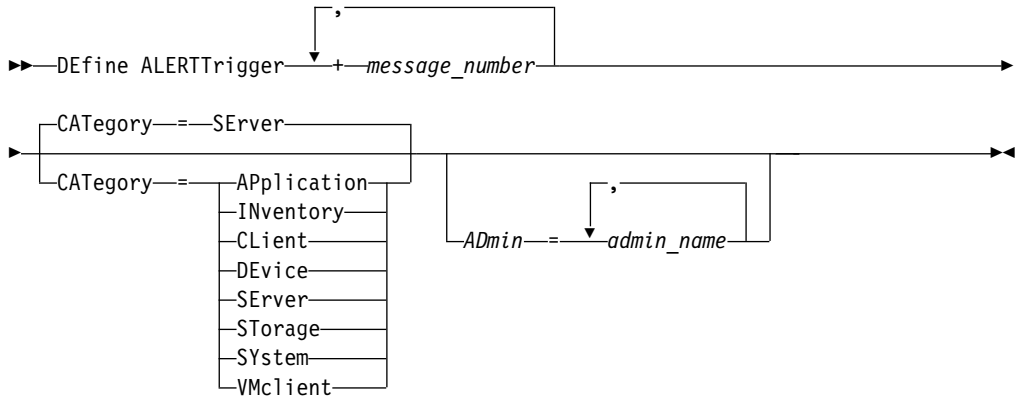
## DEFINE ALERTTRIGGER (Define an alert trigger)

Use this command to trigger an alert whenever a server issues a specific error message. You can define a message number to be an alert trigger, assign it to a category, or specify administrators who can be notified of the alert by email.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *message\_number* (Required)

Specifies the message number that you want to associate with the alert trigger. Specify multiple message numbers, which are separated by commas, and no intervening spaces. Message numbers are a maximum of eight characters in length.

#### **CATegory**

Specifies the category type for the alert, which is determined by the message types. The default value is `SERVER`.

**Note:** Changing the category of an alert trigger does not change the category of existing alerts on the server. New alerts are categorized with the new category.

Specify one of the following values:

#### **APplication**

Alert is classified as application category. For example, you can specify this category for messages that are associated with application (TDP) clients.

#### **INventory**

Alert is classified as inventory category. For example, you can specify this category for messages that are associated with the database, active log file, or archive log file.

#### **CLient**

Alert is classified as client category. For example, you can specify this category for messages that are associated with general client activities.

**DEvice**

Alert is classified as device category. For example, you can specify this category for messages that are associated with device classes, libraries, drives, or paths.

**SErver**

Alert is classified as general server category. For example, you can specify this category for messages that are associated with general server activities or events.

**STorage**

Alert is classified as storage category. For example, you can specify this category for messages that are associated with storage pools.

**SYstems**

Alert is classified under system clients category. For example, you can specify this category for messages that are associated with system backup and archive or hierarchical storage management (HSM) backup-archive clients.

**VMclient**

Alert is classified under VMclient category. For example, you can specify this category for messages that are associated with virtual machine clients.

**ADmin**

This optional parameter specifies the name of the administrator who receives email notification of this alert. The alert trigger is defined successfully even if no administrator names are specified.

**Assign two message numbers to an alert**

Issue the following command to specify that you want two message numbers to trigger an alert:

```
define alerttrigger ANR1067E,ANR1073E
```

**Assign a message number to an alert and email two administrators**

Issue the following command to specify the message numbers that you want to trigger an alert and have them sent by email to two administrators:

```
define alerttrigger ANR1067E,ANR1073E ADmin=BILL,DJADMIN
```

**Related commands**

Table 51. Commands related to **DEFINE ALERTTRIGGER**

Command	Description
"DELETE ALERTTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.
"QUERY ALERTSTATUS (Query the status of an alert)" on page 680	Displays information about alerts that have been issued on the server.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.

*Table 51. Commands related to DEFINE ALERTRIGGER (continued)*

<b>Command</b>	<b>Description</b>
"UPDATE ALERTSTATUS (Update the status of an alert)" on page 1205	Updates the status of a reported alert.

## 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 52. 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 node 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 that is running this command. The client node has the option of restoring the backup set from the server that is running this command rather than the one on which the backup set was generated.

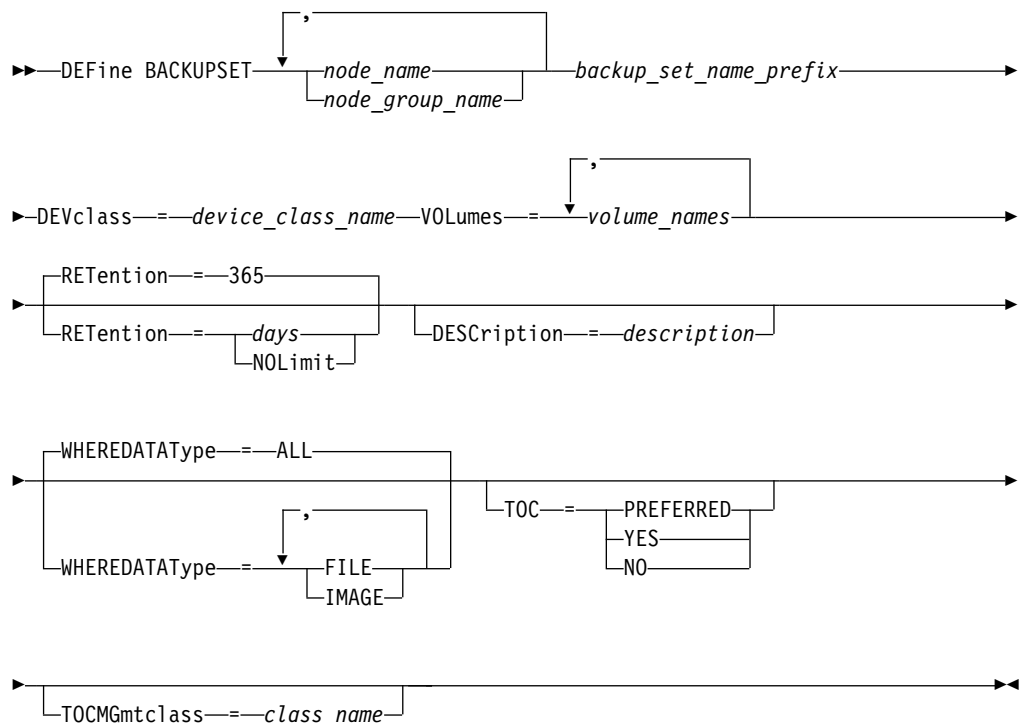
Any backup set generated on one server can be defined to another server when the servers share a common device type. The level of the server to which the backup set is being defined must be equal to or greater than the level of the server that generated the backup set.

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 is defined. If the volumes contain a backup set for a node that is not currently registered, the **DEFINE BACKUPSET** command does 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 are defined for each of the nodes by 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 that is 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 that are used to store the backup set. You can specify multiple volumes by separating the names with commas and no intervening spaces. The volumes that you specify must be available to the server that is defining the backup set.

**Note:** The volumes that you specify must be listed in the order they were created, or the **DEFINE BACKUPSET** command fails.

The server does not verify that every volume specified for a multiple-volume backup set contains part of the backup set. The first volume is always checked, and in some cases extra volumes are also checked. If these volumes are correct, the backup set is defined and all of the volumes that are 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 is not 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 fails. If this failure 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 0 - 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 must be retained on the server indefinitely.

If you specify **NOLIMIT**, Tivoli Storage Manager retains the volumes that contain 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.

**WHERE DATATYPE**

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. **ALL** is the default value.

**FILE**

Specifies that a file level backup set is to be defined. File level backup sets contain files and directories that are backed up by the backup client.

**IMAGE**

Specifies that an image backup set is to be defined. Image backup sets contain images that are created by the backup-archive client **BACKUP IMAGE** command.

**TOC**

Specifies whether a table of contents (TOC) must be created for the file level backup set when it is defined. The TOC parameter is ignored when you define 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 that is specified by the **TOCMGMTCLASS** parameter. To create a table of contents extra processing, storage pool space, and possibly a mount point during the backup set operation is required.
- If a table of contents is not saved for a backup set, you can still restore individual files or directory trees by using the backup-archive client **RESTORE BACKUPSET** command if 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 must 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 must 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 that is 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 53. 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.
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.

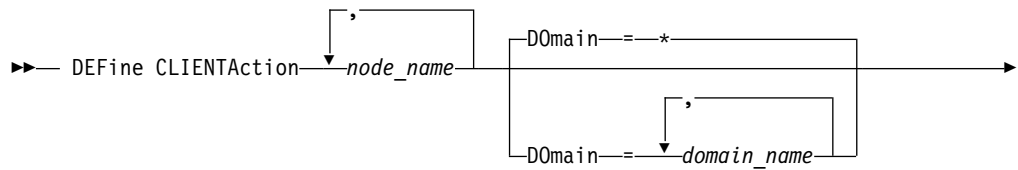
How quickly the client processes this command depends on whether the scheduling mode for the client is set to server-prompted or client-polling. The client scheduler must be started on the client workstation in order for the server to process the schedule.

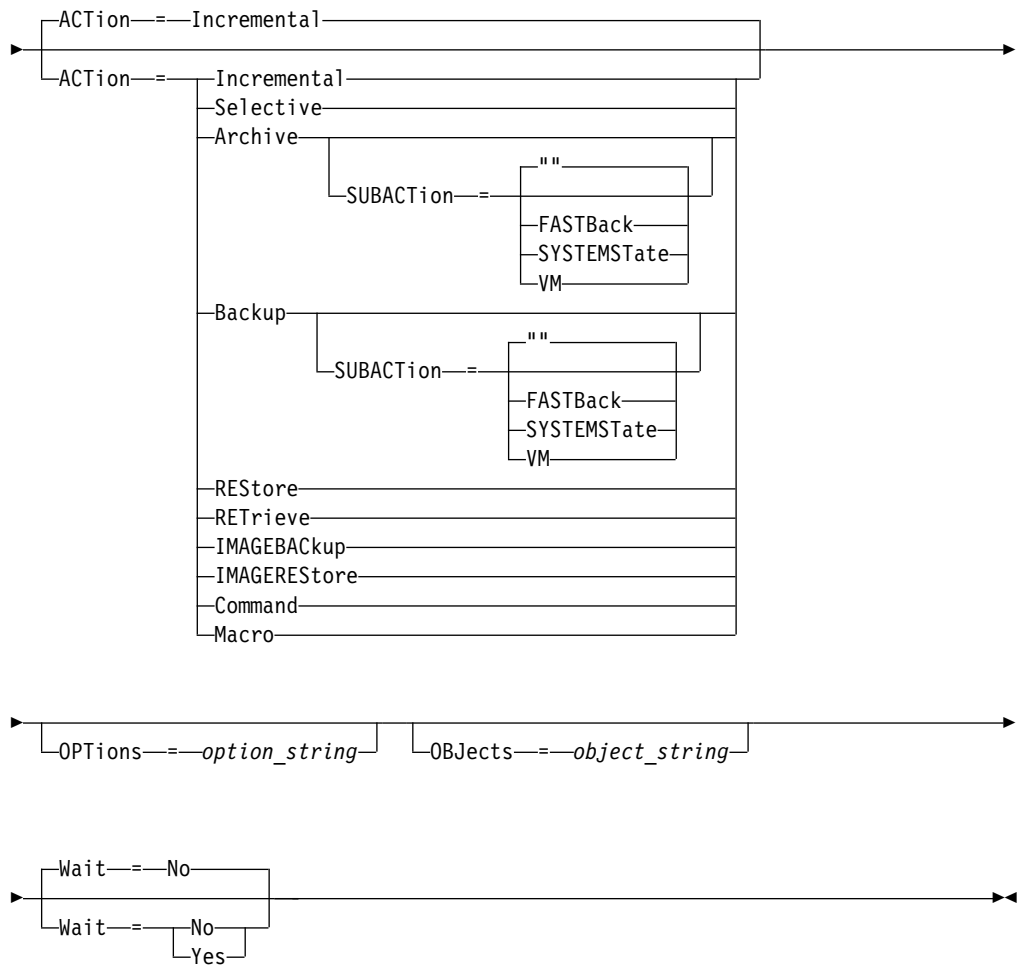
**Remember:** The start of the Tivoli Storage Manager scheduler depends on the processing of other threads in the server and other processes on the Tivoli Storage Manager server host system. The amount of time it takes to start the scheduler also depends on network traffic and how long it takes to open 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





## Parameters

### *node\_name* (Required)

Specifies the name of the client node that will process the schedule associated with the action. If you specify multiple node names, separate the names with commas; do not use intervening spaces. You can use the asterisk wildcard character to specify multiple names.

### **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 might 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.

**Backup**

Specifies that the schedule backs up 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**

Indicates that the schedule retrieves files that are specified with the OBJECTS parameter.

**Remember:** A second file that is specified acts as the retrieve destination. If you need to retrieve multiple groups of files, create a separate schedule for each group of files.

**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.

**SUBACTion**

You can specify one of the following values:

"" When a null string (two double quotes) is specified with ACTION=BACKUP the backup is an incremental.

**FASTBacK**

Specifies that a FastBack client operation that is identified by the ACTION parameter is to be scheduled for processing. The ACTION parameter must be either ARCHIVE or BACKUP.

**SYSTEMStAte**

Specifies that a client Systemstate backup is scheduled.

**VApp**

Specifies that a client vApp backup is scheduled. A vApp is a collection of pre-deployed virtual machines.

**VM** Specifies that a client VMware backup operation is scheduled.

**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 can 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 result in an error or are ignored when running the schedule from the server. For example, do not include the following options because they have no impact 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 apostrophes. Enclose individual options that contain spaces in quotation marks. 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. 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 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 /opt/file 2, /opt/gif files, and /opt/my test file, enter:  
OBJECTS="/opt/file 2" "/opt/gif files" "/opt/my test file"
- To specify /opt/test file, enter:  
OBJECTS="/opt/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 54. 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 <b>DEFINE CLIENTACTION</b> command.



## DEFINE CLIENTOPT (Define an option to an option set)

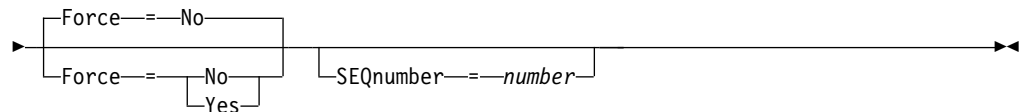
Use this command to add a client option to an option set.

### Privilege class

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

### Syntax

►►—DEFine CLIENTOpt—*option\_set\_name*—*option\_name*—*option\_value*—►►



### 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.

**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.

#### 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 140 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 PRODCLASS.

```
define clientopt winspec inclexcl "include C:\Data\...\* prodclass"
define clientopt winspec inclexcl "include 'C:\Program
Files\My Apps\...\*' prodclass"
```

**Related commands**

Table 55. 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 node to the server and sets options for that user.
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 that are 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 56. 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 or file spaces on a node 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 COLLOGrOuP—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.

### Define a collocation group

To define a node or file space collocation group named GROUP1, issue the following command:

```
define collogroup group1
```

### Related commands

Table 57. Commands related to DEFINE COLLOGROUP

Command	Description
DEFINE COLLOCMEMBER	Adds a client node or file space 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 or file space 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.

*Table 57. Commands related to DEFINE COLLOCGROUP (continued)*

<b>Command</b>	<b>Description</b>
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 COLLOCMEMBER (Define collocation group member)

Issue this command to add a client node to a collocation group or to add a file space from a node to a collocation group. A collocation group is a group of nodes or file spaces on a node 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

#### Add a node to a collocation group

```
►► DEFINE COLLOCMEMBER group_name node_name
```

### 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 to specify multiple names.

#### Add a file space from a node to a collocation group

```
►► DEFINE COLLOCMEMBER group_name node_name
```

```
►► FILESPACE file_space_name
```

NAMETYPE=SERVER  
NAMETYPE=SERVER|UNICODE|FSID  
CODETYPE=BOTH  
CODETYPE=BOTH|UNICODE|NONUNICODE

### Parameters

*group\_name*

Specifies the name of the collocation group to which you want to add a file space.

*node\_name*

Specifies the client node where the file space is located.

### **Filespace**

Specifies the *file\_space\_name* on the client node that you want to add to the collocation group. You can specify one or more file space names that are on a specific client node. If you specify multiple file space names, separate the names with commas with no intervening spaces. You can also use wildcard characters to specify multiple file space names. For example:

```
define collocmember manufacturing linux237 filespace=*_linux_fs
```

This command places all file spaces on the `linux237` node with a name that ends with `_linux_fs` into the `manufacturing` collocation group.

See the following list for tips about working with collocation groups:

- When you add members to a new collocation group, the type of the first collocation group member determines the type of the collocation group. The group can either be a node collocation group or a file space collocation group.

**Restriction:** After the collocation group type is set, it cannot be changed.

- You cannot mix collocation group member types when you add members to a collocation group (either a node group or a file space group).
- For a file space collocation group, you can add file spaces to the group. The file spaces must use the same value as the *node\_name* parameter that is specified when the collocation group is established.
- A client node can be included in multiple file space groups. However, if a node is a member of a node collocation group, it cannot be a member of a file space collocation group.
- A file space can be a member of only one file space group.

### **NAMETYPE**

Specify how you want the server to interpret the file space names that you enter. Specify this parameter when the server communicates with clients that have Unicode support. A backup-archive client with Unicode support is available only for Windows, Macintosh OS 9, Macintosh OS X, and NetWare systems. The filespace name cannot be a wildcard character when **NAMETYPE** is specified for a filespace collocation group. The default value is `SERVER`. You can specify one of the following values:

#### **SERVER**

The server uses the server 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. Whether the name can be converted depends on the characters in the names and the server code page. Conversion might fail if the string includes characters that are not available in the server code page, or if the server cannot access system conversion routines.

#### **FSID**

The server interprets the file space names by their file space IDs (FSIDs).

### **CODEType**

Specify how you want the server to interpret the file space names that you enter. Use this parameter when you use a wildcard character for the file space name. For example:

```
define collocmember production Win_3419 filespace=* codetype=unicode
```

This example command adds all file spaces from the Win\_3419 node to the production collocation group. The default is BOTH, so the file spaces are included, regardless of code page type. You can specify one of the following values:

**BOTH**

Include the file spaces, regardless of code page type.

**UNICODE**

Include file spaces that are only in Unicode.

**NONUNICODE**

Include file spaces that are not in Unicode.

## Define two collocation group members

Define two members, NODE1 and NODE2, to a collocation group, GROUP1.

```
define collocmember group1 node1,node2
```

## Define one file space group member CNTR90524, on node clifton to collocation group TSM\_alpha\_1

```
define collocmember TSM_alpha_1 clifton filespace=CNTR90524
```

## Related commands

Table 58. Commands related to DEFINE COLLOCMEMBER

Command	Description
DEFINE COLLOGROUP	Defines 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 or file space from a collocation group.
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the 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.
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.

- “DEFINE COPYGROUP (Define an archive copy group)” on page 154
- “DEFINE COPYGROUP (Define a backup copy group)” on page 149

*Table 59. 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.

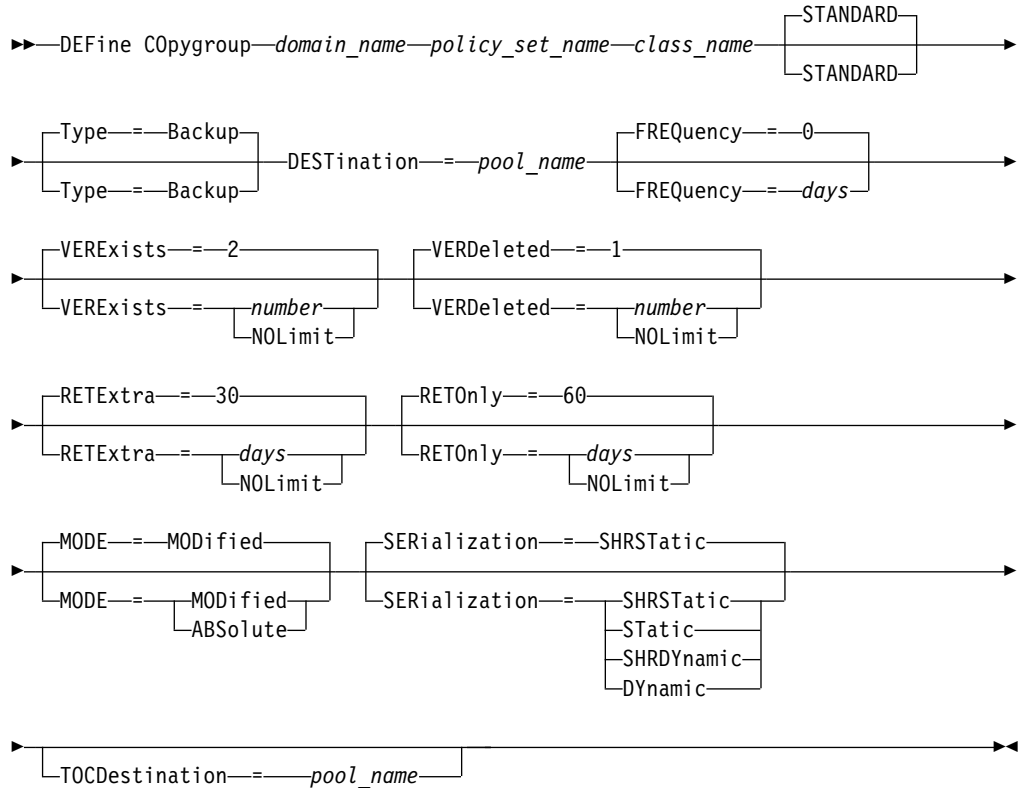
## DEFINE COPYGROUP (Define a backup copy group)

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.

**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).

**RETOnly**

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:

**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
```

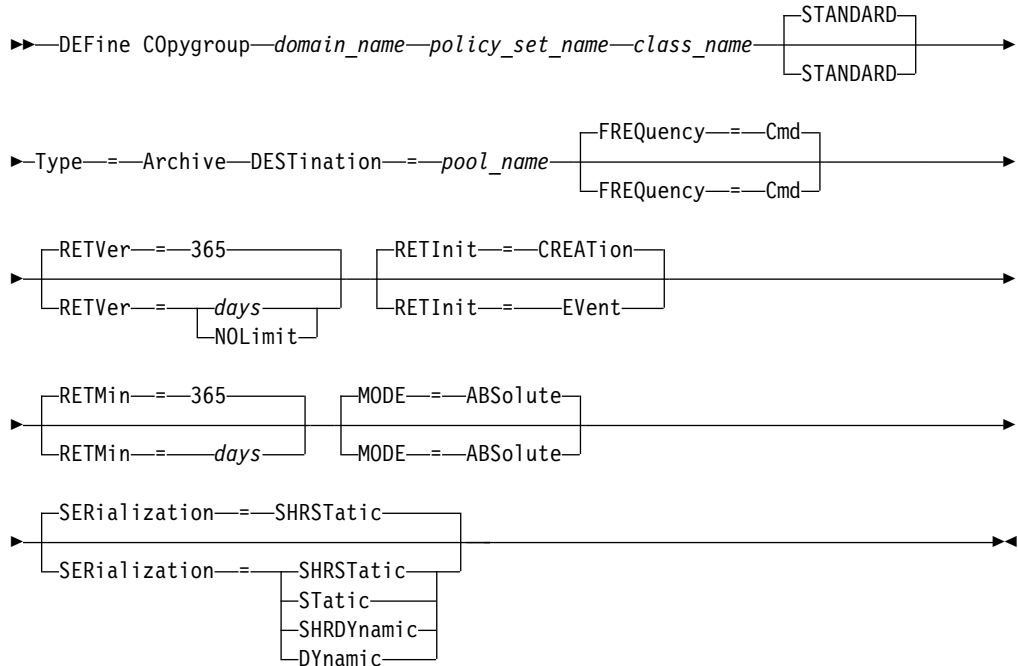
## DEFINE COPYGROUP (Define an archive copy group)

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 length of time to keep an archive copy. You can specify an integer in the range 0 - 30000.

The **RETENTIONEXTENSION** server option can affect the volume retention if the following conditions are true:

- You specify zero for the number of days
- The destination storage pool for the archive copy group is a SnapLock storage pool (RECLAMATIONTYPE=SNAPLOCK)

If the two conditions are met, retention of the volumes is defined by the value of the **RETENTIONEXTENSION** server option. The **RETENTIONEXTENSION** server option value also applies if data is copied or moved into the SnapLock storage pool by a server process such as migration, or by using the **MOVE DATA** or **MOVE NODEDATA** commands.

**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 is enabled, then the **RETVER** value is 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 later. 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 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 is initiated, but the object is not deleted while the hold is in effect.

**RETMIn**

Specifies the minimum number of days to keep an archive copy after it is 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 that is 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 that is 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 in the file because it contains some, but not all, modifications. If a file that contains a fuzzy backup is retrieved, the file might or might 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
```

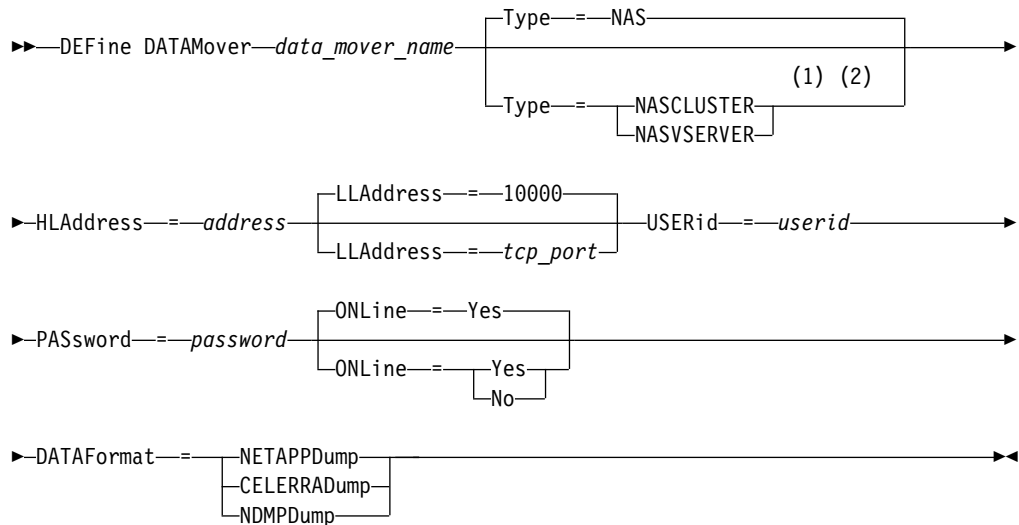
## DEFINE DATAMOVER (Define a data mover)

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. A data mover can be used to complete outboard copy operations.

### Privilege class

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

### Syntax



### Notes:

- 1 You can specify `TYPE=NASCLUSTER` and `TYPE=NASVSERVER` only on an AIX, Linux, Solaris, or Windows operating system.
- 2 You can specify `TYPE=NASCLUSTER` and `TYPE=NASVSERVER` only if `DATAFORMAT=NETAPPDUMP`.

### 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 by 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

Specifies the type of data mover. This parameter is optional. The default value is `NAS`.

#### NAS

Specifies that the data mover is a NAS file server.

#### HLAddress (Required)

Specifies either the numerical IP address or the domain name that is used to access the NAS file server.

**Tip:** To determine the numerical IP address, access the NAS file server. Then, follow the instructions in the file server documentation for obtaining the address.

**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 user ID that is configured on the NetApp file server for NDMP connections.

**Tip:** To determine the user ID, access the NAS file server. Then, follow the instructions in the file server documentation for obtaining the user ID.

**PASsword (Required)**

Specifies the password for the user ID to log on to the NAS file server.

**Tip:** To determine the password, access the NAS file server. Then, follow the instructions in the file server documentation for obtaining the password.

**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 offline.

If a library is controlled by using a path from a NAS data mover to the library, and the NAS data mover is offline, the server is not able to access the library. If the server is halted and restarted while the NAS data mover is offline, the library is not initialized.

**DATAFormat (Required)**

Specifies the data format that is used by this data mover.

**NETAPPDump**

Must be used for NetApp NAS file servers and the IBM System Storage® N Series.

**CELERRADump**

Must be used for EMC Celerra NAS file servers.

**NDMPDump**

Must be used for NAS file servers other than NetApp or EMC file servers.

**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.EXAMPLE.COM at port 10000.

```
define datamover nas1 type=nas haddress=netapp2.example.com lladdress=10000  
userid=root password=admin dataformat=netappdump
```

## Example: Define a data mover by IP address

Define a data mover for the node named NAS2. The numerical IP address for the data mover is 203.0.113.0, at port 10000. The NAS file server is not a NetApp or EMC file server.

```
define datamover nas2 type=nas hladdress=203.0.113.0 lladdress=10000
    userid=root password=admin dataformat=ndmpdump
```

## Related commands

*Table 60. Commands related to DEFINE DATAMOVER*

<b>Command</b>	<b>Description</b>
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 node 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 the use of a device.

For the most up-to-date list of supported devices and valid device class formats, see the Tivoli Storage Manager Supported Devices website:

[http://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)

**Note:** The DISK device class is defined by IBM Tivoli Storage Manager and cannot be modified with the **DEFINE DEVCLASS** command.

The following Tivoli Storage Manager device classes are ordered by device type.

- “DEFINE DEVCLASS (Define a 3590 device class)” on page 162
- “DEFINE DEVCLASS (Define a 3592 device class)” on page 166
- “DEFINE DEVCLASS (Define a 4MM device class)” on page 172
- “DEFINE DEVCLASS (Define an 8MM device class)” on page 176
- “DEFINE DEVCLASS (Define a CENTERA device class)” on page 182
- “DEFINE DEVCLASS (Define a DLT device class)” on page 184
- “DEFINE DEVCLASS (Define an ECARTRIDGE device class)” on page 190
- “DEFINE DEVCLASS (Define a FILE device class)” on page 197
- “DEFINE DEVCLASS (Define a GENERICTAPE device class)” on page 200
- “DEFINE DEVCLASS (Define an LTO device class)” on page 203
- “DEFINE DEVCLASS (Define a NAS device class)” on page 210
- “DEFINE DEVCLASS (Define a REMOVABLEFILE device class)” on page 213
- “DEFINE DEVCLASS (Define a SERVER device class)” on page 216
- “DEFINE DEVCLASS (Define a VOLSAFE device class)” on page 218

*Table 61. Commands related to DEFINE DEVCLASS*

Command	Description
BACKUP DEVCONFIG	Backs up Tivoli Storage Manager device information to a file.
DEFINE LIBRARY	Defines an automated or manual library.
DELETE DEVCLASS	Deletes 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.

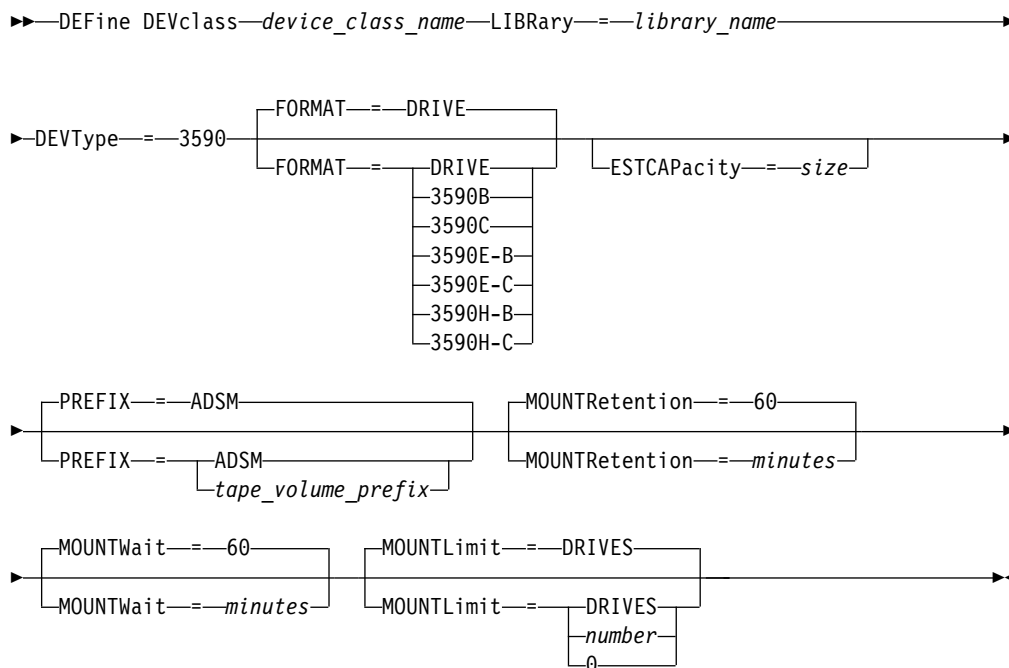
## DEFINE DEVCLASS (Define a 3590 device class)

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 data is written 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.

The following tables list the recording formats, estimated capacities, and recording format options for 3590 devices:



Table 62. 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.  <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 that contains some drives that support recording formats superior to other drives.
3590B	10.0 GB	Uncompressed (basic) format
3590C	See note 20.0 GB	Compressed format
3590E-B	10.0 GB	Uncompressed (basic) format, similar to the 3590B format
3590E-C	See note 20.0 GB	Compressed format, similar to the 3590C format
3590H-B	30.0 GB (J cartridge – standard length)  60.0 GB (K cartridge - extended length)	Uncompressed (basic) format, similar to the 3590B format
3590H-C	See note  60.0 GB (J cartridge - standard length)  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 might be greater than the listed value.

Table 63. 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 that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

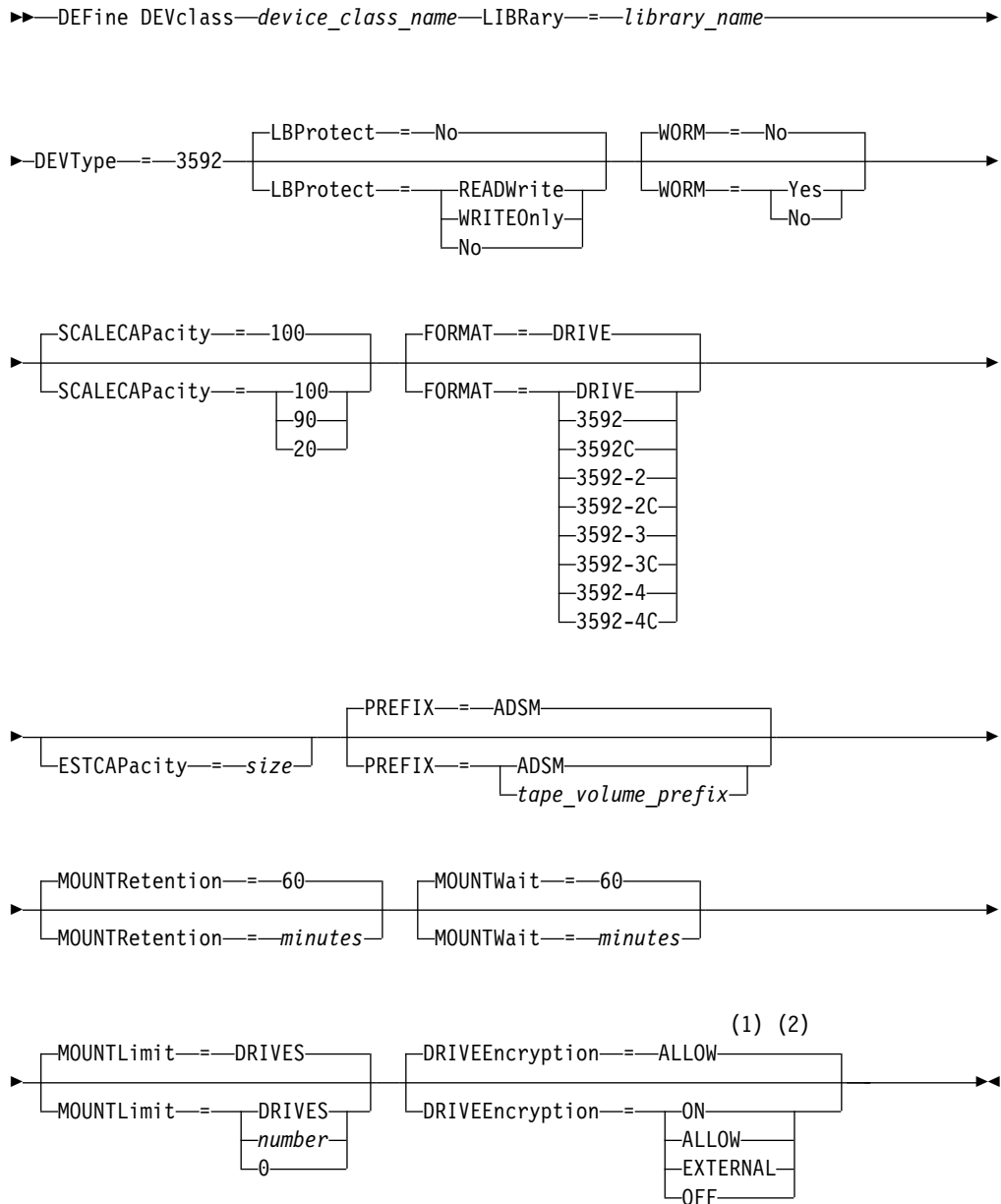
## DEFINE DEVCLASS (Define a 3592 device class)

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



### Notes:

- 1 You cannot specify both WORM=Yes and DRIVEENCRYPTION=ON.
- 2 Drive encryption is supported only for 3592 Generation 2 or later drives.

## 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 that the 3592 device type is assigned to the device class.

### **LBProtect**

Specifies whether logical block protection is used to ensure the integrity of data stored on tape. When **LBPROTECT** is set to **READWRITE** or to **WRITEONLY**, the server uses this feature of the tape drive for logical block protection and generates cyclic redundancy check (CRC) protection information for each data block written on tape. The server also validates the CRC protection information when data is read from the tape.

The default is **NO**.

The following values are possible:

#### **READWrite**

Specifies that logical block protection is enabled in the server and the tape drive for both read and write operations. Data is stored with CRC information in each block. This mode affects performance because additional processor usage is required for Tivoli Storage Manager and the tape drive to calculate and compare CRC values. The **READWRITE** value does not affect backup sets and data that is generated by the **BACKUP DB** command.

When the **LBPROTECT** parameter is set to **READWRITE**, you do not have to specify the **CRCDATA** parameter in a storage pool definition because logical block protection provides better protection against data corruption.

#### **WRITEOnly**

Specifies that logical block protection is enabled in the server and the tape drive for write operations only. Data is stored containing CRC information in each block. For read operations, the server and the tape drive do not validate the CRC. This mode affects performance because additional processor usage is required for Tivoli Storage Manager to generate the CRC and for the tape drive to calculate and compare CRC values for write operations. The **WRITEONLY** value does not affect backup sets and data that are generated by the **BACKUP DB** command.

**No** Specifies that logical block protection is not enabled in the server and the tape drive for read and write operations. However, the server enables logical block protection on write operations for a filling volume that already has data with logical block protection.

**Restriction:** Logical block protection is supported only on IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.

See Technote 1634851 at <http://www.ibm.com/support/docview.wss?uid=swg21634851> for an explanation about when to use the **LBProtect** parameter.

#### **WORM**

Specifies whether the drives use WORM (write once, read many) media. This parameter is optional. The default is **No**. The field can contain one of the following values:

##### **Yes**

Specifies that the drives use WORM media.

**No** Specifies that the drives do not use WORM media.

##### **Remember:**

1. To use 3592 WORM support in 3584 libraries, you must specify the WORM parameter. The server distinguishes between WORM and non-WORM scratch volumes. However, to use 3592 WORM support in 349X libraries, you also must set the WORMSCRATCHCATEGORY on the **DEFINE LIBRARY** command. For details, see “DEFINE LIBRARY (Define a library)” on page 232.
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.

#### **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 takes effect only when data is first written to a volume. Any updates to the device class for scale capacity do not affect volumes that already have data that is written to them until the volume is returned to scratch status.

#### **FORMAT**

Specifies the recording format to be used when data is written 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.

The following table lists the recording formats, estimated capacities, and recording format options for 3592 devices:

Table 64. 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.  <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 that contains some drives that support recording formats superior to other drives.
3592	300 GB	Uncompressed (basic) format
3592C	See note 900 GB	Compressed format
3592-2	500 GB 700 GB	Uncompressed (basic) format JA tapes Uncompressed (basic) format JB tapes
3592-2C	1.5 TB 2.1 TB	Compressed format JA tapes Compressed format JB tapes
3592-3	640 GB 1 TB	Uncompressed (basic) format JA tapes Uncompressed (basic) format JB tapes
3592-3C	1.9 TB 3 TB	Compressed format JA tapes Compressed format JB tapes
3592-4	400 GB 1.5 TB 3.1 TB	Uncompressed (basic) format JK tapes Uncompressed (basic) format JB tapes Uncompressed (basic) format JC tape
3592-4C	1.2 TB 4.4 TB 9.4 TB	Compressed format JK tapes Compressed format JB tapes Compressed format JC tapes

**Note:** If this format uses the tape drive hardware compression feature, depending on the effectiveness of compression, the actual capacity might be different from the listed value.

**Important:** For optimal performance, avoid mixing different generations of drives in a single SCSI library.

Special configurations are also required for mixing different generations of 3592 drives in 349x and ACSLS libraries.

#### **ESTCAPacity**

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

#### **DRIVEEncryption**

Specifies whether drive encryption is allowed. This parameter is optional. The default is ALLOW.

**ON** Specifies that Tivoli Storage Manager is the key manager for drive encryption and allows drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes—for example, back up 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 is not allowed and backup operations fail.

#### **ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is allowed if either the library or system 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 off encryption. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns off encryption.

#### **OFF**

Specifies that drive encryption is not allowed. If you enable either the library or system method of encryption, backups fail. If you enable the application method, Tivoli Storage Manager disables encryption and backups are attempted.

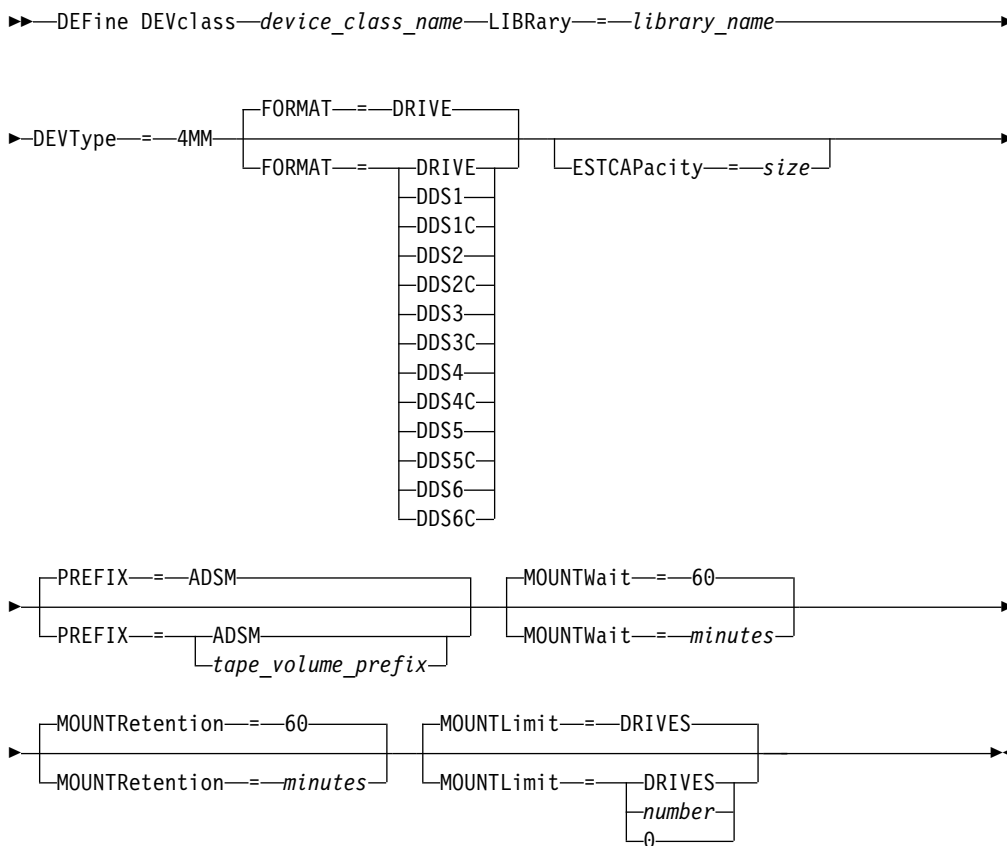
## DEFINE DEVCLASS (Define a 4MM device class)

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 data is written 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.

The following table lists the recording formats and estimated capacities for 4 mm devices:

*Table 65. Recording formats and default estimated capacities for 4 mm tapes*

<b>Format</b>	<b>Estimated Capacity</b>	<b>Description</b>
DRIVE	–	The server selects the highest format that is supported by the drive on which a volume is mounted.  <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 that contains some drives that support recording formats superior to other drives.
DDS1	2.6 GB (60 meter) 4.0 GB (90 meter)	Uncompressed format, applies only to 60-meter and 90-meter tapes
DDS1C	See note 1.3 GB (60 meter) 2.0 GB (90 meter)	Compressed format, applies only to 60-meter and 90-meter tapes
DDS2	4.0 GB	Uncompressed format, applies only to 120-meter tapes
DDS2C	See note 8.0 GB	Compressed format, applies only to 120-meter tapes
DDS3	12.0 GB	Uncompressed format, applies only to 125-meter tapes
DDS3C	See note 24.0 GB	Compressed format, applies only to 125-meter tapes
DDS4	20.0 GB	Uncompressed format, applies only to 150-meter tapes
DDS4C	See note 40.0 GB	Compressed format, applies only to 150-meter tapes
DDS5	36 GB	Uncompressed format, when using DAT 72 media
DDS5C	See note 72 GB	Compressed format, when using DAT 72 media
DDS6	80 GB	Uncompressed format, when using DAT 160 media
DDS6C	See note 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 might be greater than the listed value.

#### **ESTCAPacity**

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

For more information about the default estimated capacity for 4 mm tapes, see Table 65 on page 173

#### **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 is **ADSM**. The maximum length of this prefix is 8 characters.

If you have a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

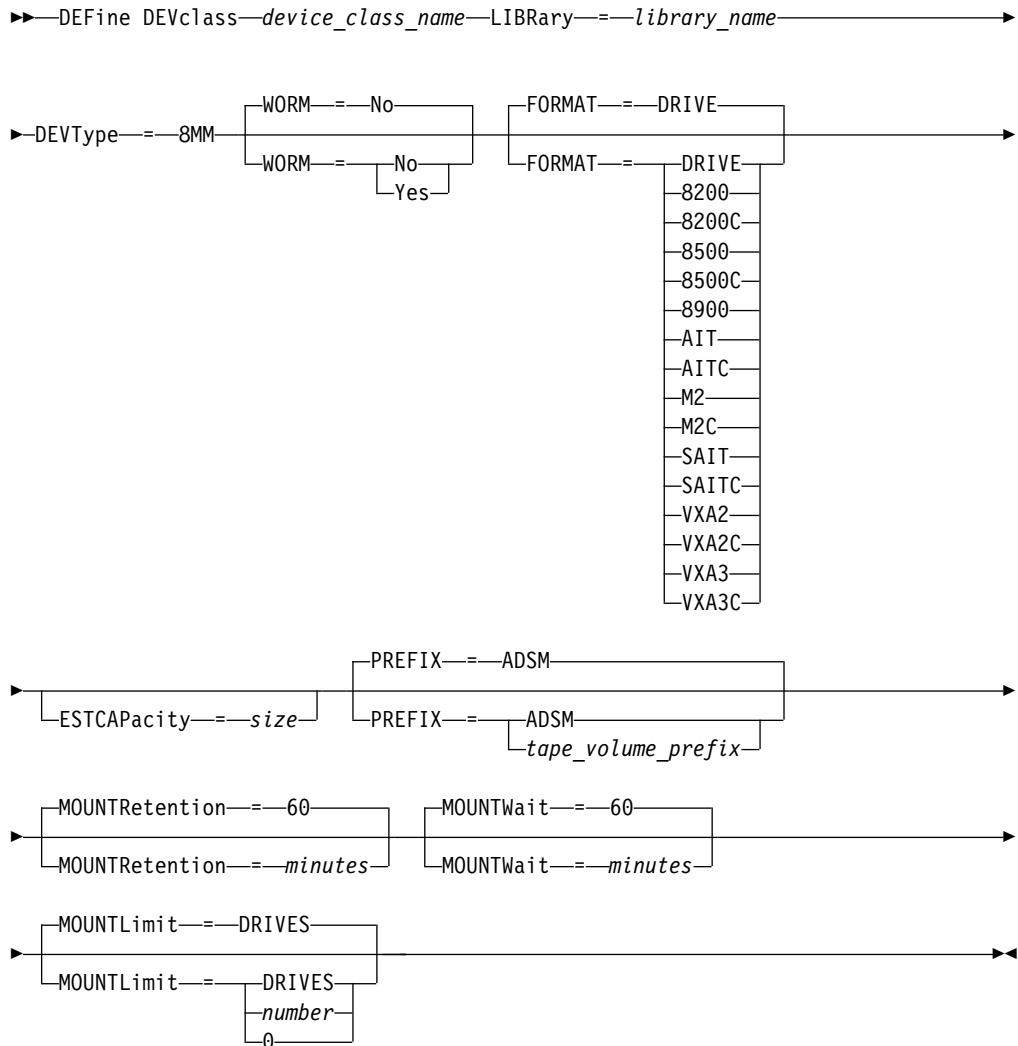
## DEFINE DEVCLASS (Define an 8MM device class)

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 use WORM (write once, read many) media. This parameter is optional. The default is **No**. The field can contain one of the following values:

**Yes**

Specifies that the drives use WORM media.

**No** Specifies that the drives do not use WORM media.

**Note:** If you select Yes, the only options available for the FORMAT parameter are:

- DRIVE
- AIT
- AITC

**FORMAT**

Specifies the recording format to be used when data is written 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.

The following table lists the recording formats and estimated capacities for 8 mm devices:

*Table 66. Recording format and default estimated capacity for 8 mm tape*

Format	Estimated Capacity	Description
DRIVE	–	The server selects the highest format that is supported by the drive on which a volume is mounted.  <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 that contains 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 3.5 GB 4.6 GB	Compressed format, using standard 112-meter tape cartridges

Table 66. Recording format and default estimated capacity for 8 mm tape (continued)

Format		
Medium Type	Estimated Capacity	Description
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 66. 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 might 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 that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

For more information about the default estimated capacity for 8 mm tapes, see Table 66 on page 177.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

### **Example: Define an 8 mm device class**

Define a device class that is 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
```

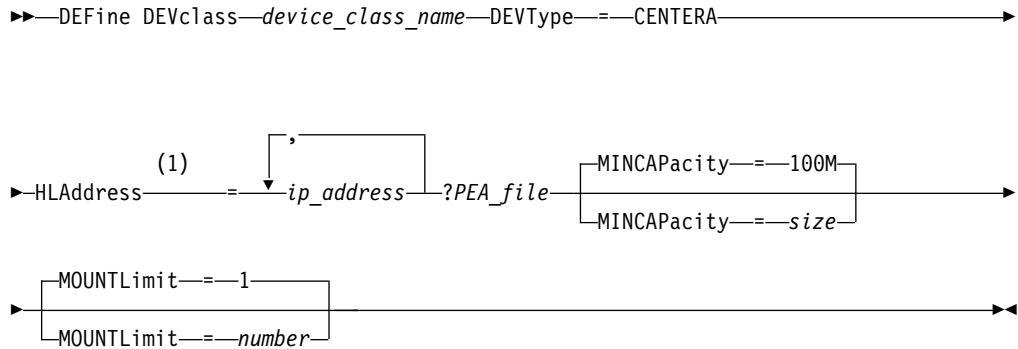
## DEFINE DEVCLASS (Define a CENTERA device class)

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:

- 1 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 that belong 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 with 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 attempts a connection by using each IP address that is 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 that runs the server. Separate the PEA file name and path from the IP address with 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 server uses the PEA file that is specified in the device class HLADDRESS parameter that was first used to open the Centera storage device.

**Tips:**

1. The server does not include a PEA file during installation. If you do not create a PEA file, the 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 with the command-line interface that is 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 with the syntax `CENTERA_PEA_LOCATION=filePath_fileName`. The PEA file name and path that is specified with this environment variable apply to all Centera clusters. If you use this variable, you do not have to specify the PEA file name and path with the HLADDRESS parameter.

**MINCAPacity**

Specifies the minimum size for Centera volumes that are assigned to a storage pool in this device class. This value represents the minimum amount of data that is stored on a Centera volume before the server marks it full. Centera volumes continue to accept data until the minimum amount of data is stored. This parameter is optional.

Specify this value as an integer followed by K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The default value is 100 MB (MINCAPACITY=100M). The minimum value that is allowed is 1 MB (MINCAPACITY=1M). The maximum value that is 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 that are assigned to the same Centera device must not exceed the maximum number of sessions that are allowed by Centera.

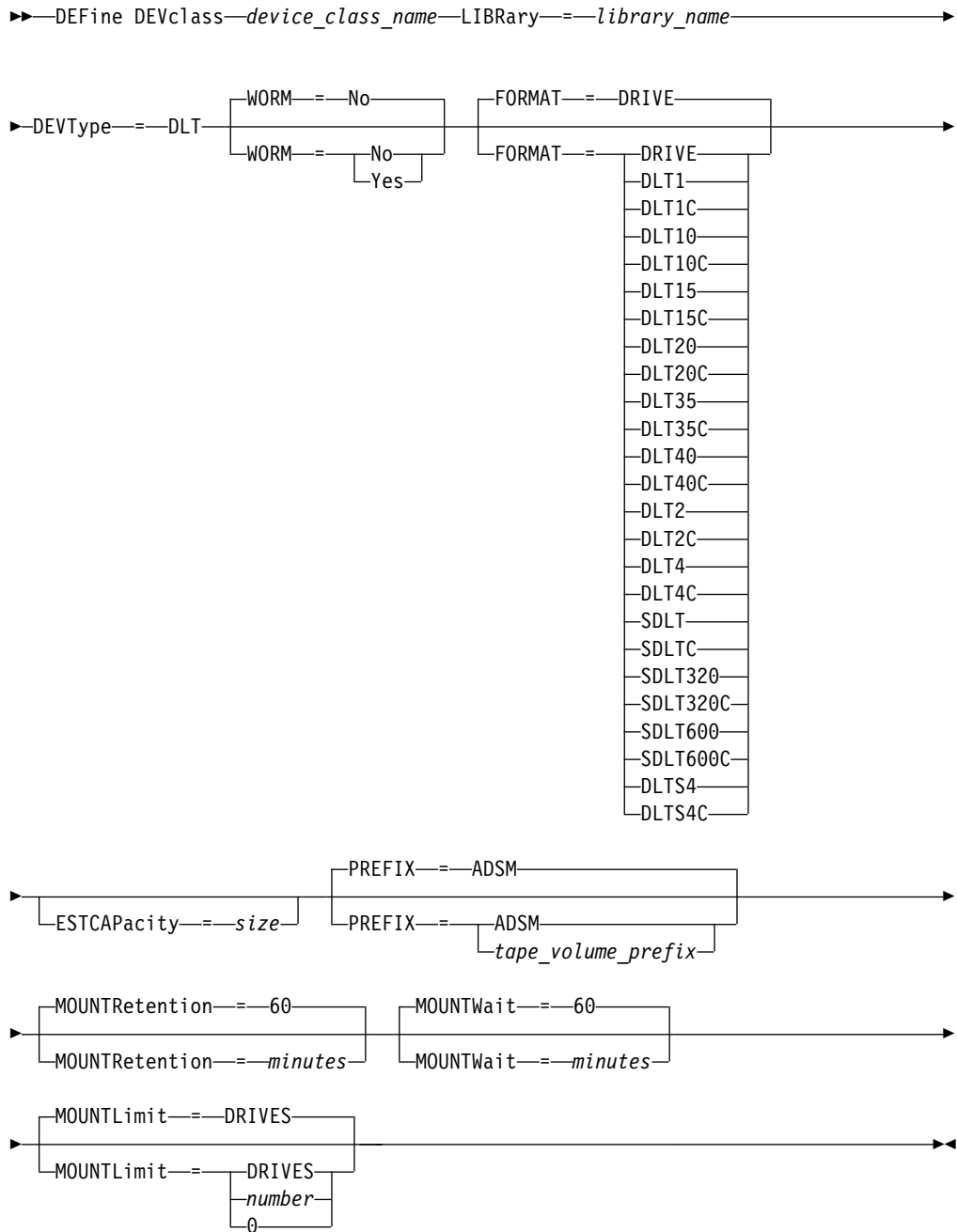
## DEFINE DEVCLASS (Define a DLT device class)

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 use WORM (write once, read many) media. This parameter is optional. The default is **No**. The field can contain one of the following values:

#### Yes

Specifies that the drives use WORM media.

**No** Specifies that the drives do not 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.

### FORMAT

Specifies the recording format to be used when data is written 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.

The following table lists the recording formats and estimated capacities for DLT devices:

Table 67. 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.  <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 that contains some drives that support recording formats superior to other drives.
DLT1	40.0 GB	Uncompressed format, using only CompacTape III cartridges  Valid with DLT4000, DLT7000, and DLT8000 drives
DLT1C	See note 1 on page 187.  80.0 GB	Compressed format, using only CompacTape III cartridges  Valid with DLT4000, DLT7000, and DLT8000 drives

Table 67. Recording format and default estimated capacity for DLT (continued)

Format	Estimated Capacity	Description
DLT10	10.0 GB	Uncompressed format, using only CompacTape III cartridges Valid with DLT4000, DLT7000, and DLT8000 drives
DLT10C	See note 1 on page 187. 20.0 GB	Compressed format, using only CompacTape III cartridges 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 187. 30.0 GB	Compressed format, using only CompacTape IIIxt cartridges 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 187. 40.0 GB	Compressed format, using only CompacTape IV cartridges 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 187. 70.0 GB	Compressed format, using only CompacTape IV cartridges 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 187. 80.0 GB	Compressed format, using CompacTape IV cartridges Valid with a DLT8000 drive
DLT2	80.0 GB	Uncompressed format, using Quantum DLT tape VS1 media
DLT2C	See note 1 on page 187. 160.0 GB	Compressed format, using Quantum DLT tape VS1 media
DLT4	160.0 GB	Uncompressed format, using Quantum DLTtape VS1 cartridges. Valid with Quantum DLT-V4 drive



Table 67. Recording format and default estimated capacity for DLT (continued)

Format	Estimated Capacity	Description
DLT4C	See note 1. 320.0 GB	Compressed format, using Quantum DLTtape VS1 cartridges.  Valid with Quantum DLT-V4 drive
SDLT See note 2.	100.0 GB	Uncompressed format, using Super DLT Tape 1 cartridges  Valid with a Super DLT drive
SDLTC See note 2.	See note 1. 200.0 GB	Compressed format, using Super DLT Tape 1 cartridges  Valid with a Super DLT drive
SDLT320 See note 2.	160.0 GB	Uncompressed format, using Quantum SDLT I media  Valid with a Super DLT drive
SDLT320C See note 2.	See note 1. 320.0 GB	Compressed format, using Quantum SDLT I media  Valid with a Super DLT drive
SDLT600	300.0 GB	Uncompressed format, using SuperDLTtape-II media  Valid with a Super DLT drive
SDLT600C	See note 1. 600.0 GB	Compressed format, using SuperDLTtape-II media  Valid with a Super DLT drive
DLTS4	800 GB	Uncompressed format, using Quantum DLT S4 media.  Valid with a DLT-S4 drive
DLTS4C	See note 1. 1.6 TB	Compressed format, using Quantum DLT S4 media.  Valid with a DLT-S4 drive

**Note:**

1. Depending on the effectiveness of compression, the actual capacity might 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 that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

For more information about estimated capacities, see Table 67 on page 185.

**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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

## DEFINE DEVCLASS (Define an ECARTRIDGE device class)

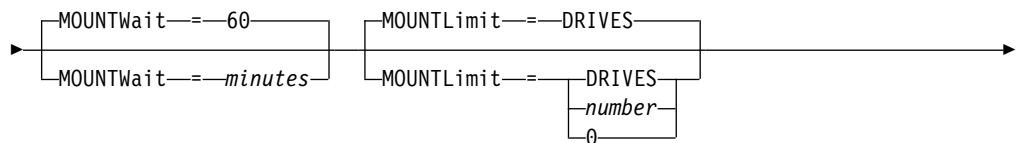
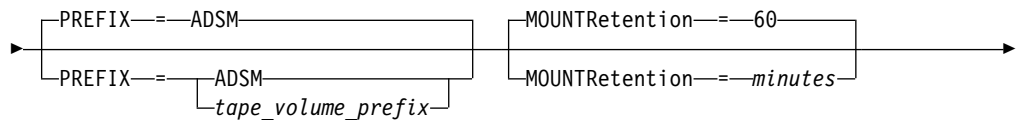
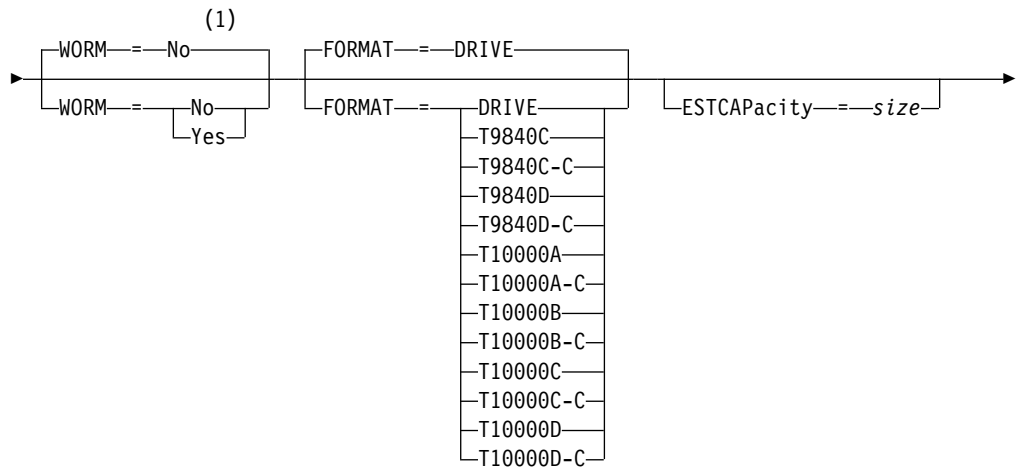
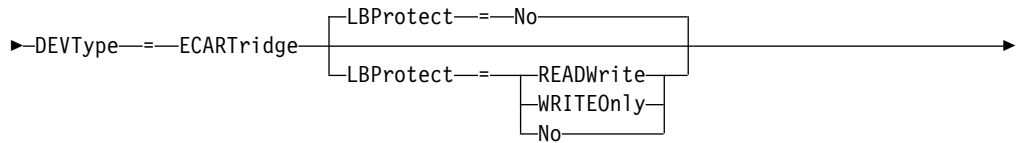
Use the ECARTRIDGE device class when you are using StorageTek drives such as the StorageTek T9840 or T10000.

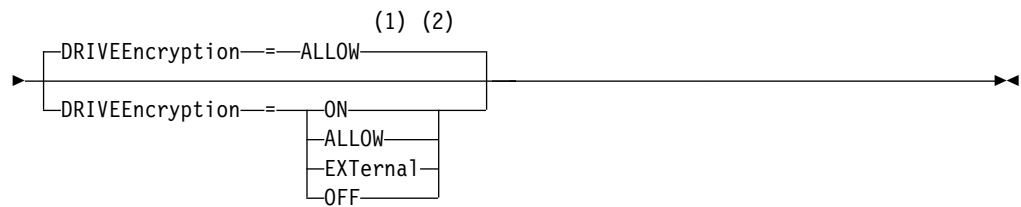
### Privilege class

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 DRIVEENCRYPTION=ON.
- 2 You can use drive encryption only for Oracle StorageTek T10000B drives with a format value of DRIVE, T10000B, or T10000B-C, for Oracle StorageTek T10000C drives with a format value of DRIVE, T10000C or T10000C-C, and for Oracle StorageTek T10000D drives with a format value of DRIVE, T10000D and T10000D-C.

**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.

**LBProtect**

Specifies whether logical block protection is used to ensure the integrity of data stored on tape. When **LBPROTECT** is set to READWRITE or to WRITEONLY, the server uses this feature of the tape drive for logical block protection and generates cyclic redundancy check (CRC) protection information for each data block written on tape. The server also validates the CRC protection information when data is read from the tape.

The default is NO.

The following values are possible:

**READWrite**

Specifies that logical block protection is enabled in the server and the tape drive for both read and write operations. Data is stored with CRC information in each block. This mode affects performance because additional processor usage is required for Tivoli Storage Manager and the tape drive to calculate and compare CRC values. The READWRITE value does not affect backup sets and data that is generated by the **BACKUP DB** command.

When the **LBPROTECT** parameter is set to READWRITE, you do not have to specify the **CRCDATA** parameter in a storage pool definition because logical block protection provides better protection against data corruption.

**WRITEOnly**

Specifies that logical block protection is enabled in the server and the tape

drive for write operations only. Data is stored containing CRC information in each block. For read operations, the server and the tape drive do not validate the CRC. This mode affects performance because additional processor usage is required for Tivoli Storage Manager to generate the CRC and for the tape drive to calculate and compare CRC values for write operations. The WRITEONLY value does not affect backup sets and data that are generated by the **BACKUP DB** command.

**No** Specifies that logical block protection is not enabled in the server and the tape drive for read and write operations. However, the server enables logical block protection on write operations for a filling volume that already has data with logical block protection.

**Restriction:** Logical block protection is supported only on Oracle StorageTek T1000C and Oracle StorageTek T1000D drives.

#### **WORM**

Specifies whether the drives use WORM (write once, read many) media. This parameter is optional. The default is **No**. The field can contain one of the following values:

##### **Yes**

Specifies that the drives use WORM media.

**No** Specifies that the drives do not use WORM media.

**Restriction:** If you select Yes, the only options that are available for the FORMAT parameter are:

- DRIVE
- T9840C
- T9840C-C
- T9840D
- T9840D-C
- T1000A
- T1000A-C
- T1000B
- T1000B-C
- T1000C
- T1000C-C
- T1000D
- T1000D-C

#### **FORMAT**

Specifies the recording format to be used when data is written 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.

**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 that is 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 68. 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.  <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 that contains some drives that support recording formats superior to other drives.
T9840C	40 GB	Uncompressed T9840C format, using a StorageTek 9840 cartridge
T9840C-C	80 GB	Compressed T9840C format, using a StorageTek 9840 cartridge
T9840D	75 GB	Uncompressed T9840D format, using a StorageTek 9840 cartridge
T9840D-C	150 GB	Compressed T9840D format, using a StorageTek 9840 cartridge
T10000A	500 GB	Uncompressed T10000A format, using a StorageTek T10000 cartridge
T10000A-C	1 TB	Compressed T10000A format, using a StorageTek T10000 cartridge
T10000B	1 TB	Uncompressed T10000B format, using an Oracle StorageTek T10000 cartridge
T10000B-C	2 TB	Compressed T10000B format, using an Oracle StorageTek T10000 cartridge
T10000C	5 TB	Uncompressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge
T10000C-C	10 TB	Compressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge
T10000D	8 TB	Uncompressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge
T10000D-C	15 TB	Compressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge

**Notes:**

- Some formats use a tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity might 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. T10000C drives can read, but cannot write, the T10000A and T10000B formats. T10000D drives can read, but cannot write, the T10000A, T10000B, and T10000C formats.

**ESTCAPacity**

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

**DRIVEEncryption**

Specifies whether drive encryption is allowed. This parameter is optional. The default is ALLOW.

**Restrictions:**

1. You can use drive encryption only for the following drives:
    - Oracle StorageTek T10000B drives that have a format value of DRIVE, T10000B, or T10000B-C
    - Oracle StorageTek T10000C drives that have a format value of DRIVE, T10000C, or T10000C-C
    - Oracle StorageTek T10000D drives that have a format value of DRIVE, T10000D, or T10000D-C
  2. You cannot specify Tivoli Storage Manager as the key manager for drive encryption of write once, read many (WORM) media. You cannot specify both WORM=Yes and DRIVEENCRYPTION=ON.
  3. If encryption is enabled for a device class, and the device class is associated with a storage pool, the storage pool should not share a scratch pool with other device classes that cannot be encrypted. If a tape is encrypted, and you plan to use it on a drive that cannot be encrypted, you must manually relabel the tape before it can be used on that drive.
- ON** Specifies that Tivoli Storage Manager is the key manager for drive encryption and allows drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes are not encrypted. For example, back up sets, export volumes, and database

backup volumes are not encrypted.) If you specify ON and you enable another method of encryption, drive encryption is not allowed and backup operations fail.

**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is allowed 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 off encryption. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns off encryption.

**OFF**

Specifies that drive encryption is not allowed. If you enable another method of encryption, backups fail. If you enable the application method, Tivoli Storage Manager disables encryption and backups are attempted.

## DEFINE DEVCLASS (Define a FILE device class)

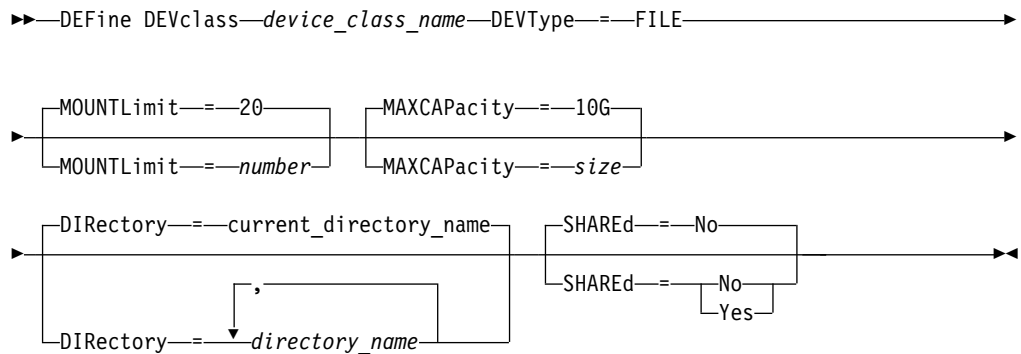
Use the FILE device class when you are using files on magnetic disk storage as volumes that store data sequentially (as on tape).

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 must 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 0 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 fails.

#### `MAXCAPacity`

Specifies the maximum size of any data storage files that are 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 10 GB (`MAXCAPACITY=10G`). The value that is specified must 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), G (gigabytes), or T (terabytes). The minimum size is 1 MB (**MAXCAPACITY=1M**). If you are defining a FILE device class for database-backup volumes, specify a value for **MAXCAPACITY** that is appropriate for the size of the database and that minimizes the number of database volumes.

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 that are used in this device class. Enclose the entire list of directories within quotation marks, and use commas to separate individual directory names. Special characters (for example, blank spaces) are allowed 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 that are specified with DIRECTORY parameter must point to the directory or directories on the NetApp SnapLock volumes.

While the command is processed, the server expands the specified directory name or names into their fully qualified forms, starting from the root directory.

If the server must 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 that is 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 262.

**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 that remains in each directory. If you specify multiple directories for a device class and the directories are in the same file system, the server calculates space by adding values that represent the space that remains in each directory. These space calculations are 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 is shared between the server and one or more storage agents. To prepare for sharing, a library is 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 Support Portal.

#### **Example: Define a FILE device class with multiple directories**

Define a device class that specifies multiple directories.

```
define devclass multidir devtype=file
    directory=/opt/xyz,/opt/abc,/opt/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)

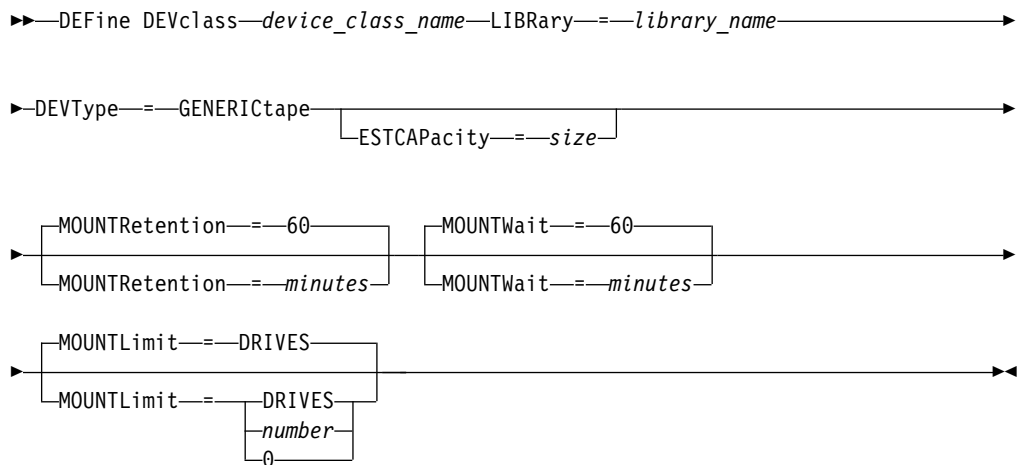
Use the GENERICTAPE device class for tape drives that are supported by operating system device drivers.

When you use 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 you define 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 that can be 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 that are supported by the operating system's tape device driver.

The server recognizes that the media can be removed and that more 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 that are 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 that is being used.

You must specify this value as an integer followed by one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

#### **MOUNTRetention**

Specifies the number of minutes that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.



## DEFINE DEVCLASS (Define an LTO device class)

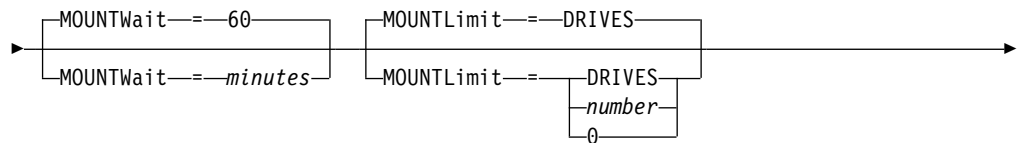
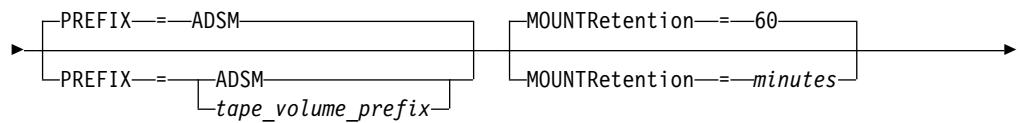
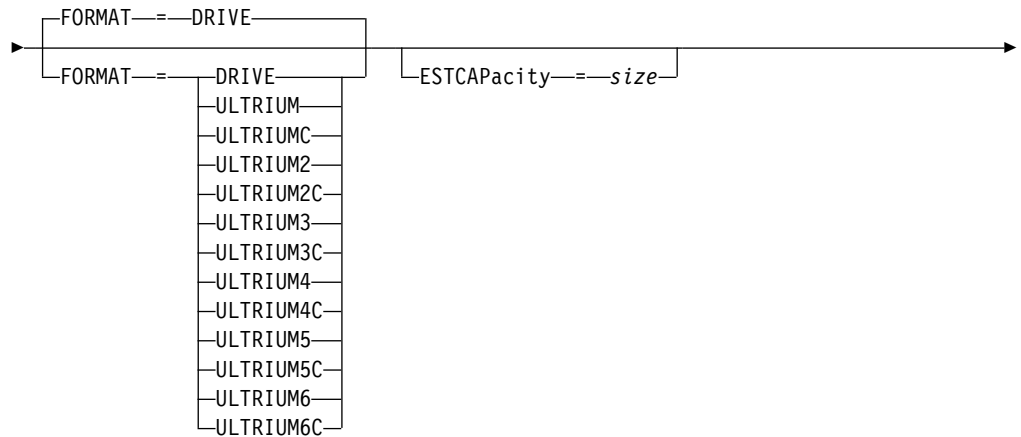
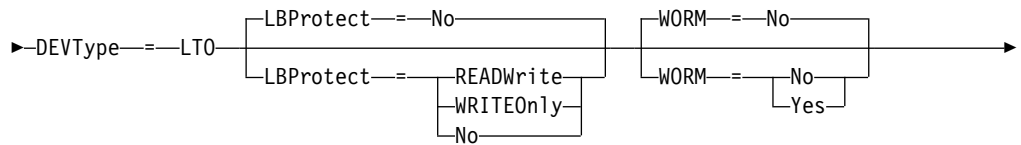
Use the LTO device class when you are using LTO tape devices.

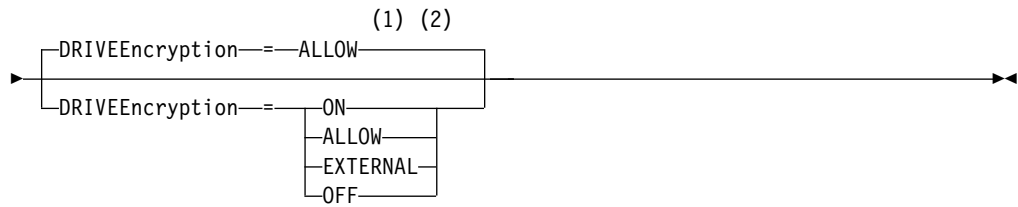
### Privilege class

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 DRIVEENCRYPTION=ON.
- 2 Drive encryption is supported only for Ultrium 4, Ultrium 5, and Ultrium 6 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.

**LBProtect**

Specifies whether logical block protection is used to ensure the integrity of data stored on tape. When **LBPROTECT** is set to READWRITE or to WRITEONLY, the server uses this feature of the tape drive for logical block protection and generates cyclic redundancy check (CRC) protection information for each data block written on tape. The server also validates the CRC protection information when data is read from the tape.

The default is NO.

The following values are possible:

**READWrite**

Specifies that logical block protection is enabled in the server and the tape drive for both read and write operations. Data is stored with CRC information in each block. This mode affects performance because additional processor usage is required for Tivoli Storage Manager and the tape drive to calculate and compare CRC values. The READWRITE value does not affect backup sets and data that is generated by the **BACKUP DB** command.

When the **LBPROTECT** parameter is set to READWRITE, you do not have to specify the **CRCDATA** parameter in a storage pool definition because logical block protection provides better protection against data corruption.

**WRITEOnly**

Specifies that logical block protection is enabled in the server and the tape drive for write operations only. Data is stored containing CRC information in each block. For read operations, the server and the tape drive do not validate the CRC. This mode affects performance because additional processor usage is required for Tivoli Storage Manager to generate the CRC

and for the tape drive to calculate and compare CRC values for write operations. The WRITEONLY value does not affect backup sets and data that are generated by the **BACKUP DB** command.

**No** Specifies that logical block protection is not enabled in the server and the tape drive for read and write operations. However, the server enables logical block protection on write operations for a filling volume that already has data with logical block protection.

**Restriction:** Logical block protection is supported only on IBM LTO5 and supported LTO6 drives.

#### WORM

Specifies whether the drives use WORM (write once, read many) media. This parameter is optional. The default is **No**. The field can contain one of the following values:

##### Yes

Specifies that the drives use WORM media.

**No** Specifies that the drives do not 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.)

#### FORMAT

Specifies the recording format to be used when data is written 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.

When migrating all drives from Ultrium to Ultrium 2 devices:

- Delete all existing Ultrium drive definitions and the paths that are 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.

Table 69. Read - write capabilities for different generations of LTO drives

Drives	Generation 1 media	Generation 2 media	Generation 3 media	Generation 4 media	Generation 5 media	Generation 6 media
Generation 1	Read and write	n/a	n/a	n/a	n/a	n/a
Generation 2	Read and write	Read and write	n/a	n/a	n/a	n/a
Generation 3 <sup>1</sup>	Read only	Read and write	Read and write	n/a	n/a	n/a
Generation 4 <sup>2</sup>	n/a	Read only	Read and write	Read and write	n/a	n/a
Generation 5 <sup>3</sup>	n/a	n/a	Read only	Read and write	Read and write	n/a

Table 69. Read - write capabilities for different generations of LTO drives (continued)

Drives	Generation 1 media	Generation 2 media	Generation 3 media	Generation 4 media	Generation 5 media	Generation 6 media
Generation 6 <sup>4</sup>	n/a	n/a	n/a	Read only	Read and write	Read and write
<p><sup>1</sup> In a library with a Generation 3 drive, all Generation 1 scratch volumes must be checked out, and all Generation 1 storage pool volumes must be updated to read-only.</p> <p><sup>2</sup> In a library with a Generation 4 drive, all Generation 2 scratch volumes must be checked out, and all Generation 2 storage pool volumes must be updated to read-only.</p> <p><sup>3</sup> In a library with a Generation 5 drive, all Generation 3 scratch volumes must be checked out, and all Generation 3 storage pool volumes must be updated to read-only.</p> <p><sup>4</sup> In a library with a Generation 6 drive, all Generation 4 scratch volumes must be checked out, and all Generation 4 storage pool volumes must be updated to read-only.</p>						

The following table lists the recording formats and estimated capacities for LTO devices:

Table 70. 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.  <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 that contains some drives that support recording formats superior to other drives.
ULTRIUM	100 GB	Uncompressed format, using Ultrium cartridges
ULTRIUMC	See note 200 GB	Compressed format, using Ultrium cartridges
ULTRIUM2	200 GB	Uncompressed (standard) format, using Ultrium 2 cartridges
ULTRIUM2C	See note 400 GB	Compressed format, using Ultrium 2 cartridges
ULTRIUM3	400 GB	Uncompressed (standard) format, using Ultrium 3 cartridges
ULTRIUM3C	See note 800 GB	Compressed format, using Ultrium 3 cartridges
ULTRIUM4	800 GB	Uncompressed (standard) format, using Ultrium 4 cartridges
ULTRIUM4C	See note 1.6 TB	Compressed format, using Ultrium 4 cartridges
ULTRIUM5	1.5 TB	Uncompressed (standard) format, using Ultrium 5 cartridges

Table 70. Recording format and default estimated capacity for LTO (continued)

Format	Estimated capacity	Description
ULTRIUM5C	See note 3.0 TB	Compressed format, using Ultrium 5 cartridges
ULTRIUM6	2.5 TB	Uncompressed (standard) format, using Ultrium 6 cartridges
ULTRIUM6C	See note 6.25 TB	Compressed format, using Ultrium 6 cartridges

**Note:** If this format uses the tape-drive hardware-compression feature, depending on the effectiveness of compression, the actual capacity might be greater than the listed value.

### ESTCAPacity

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

For more information about estimated capacities, see Table 70 on page 206.

### 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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

#### **DRIVEEncryption**

Specifies whether drive encryption is allowed. This parameter is optional. The default is ALLOW. Drive encryption is supported only for Ultrium 4, Ultrium 5, and Ultrium 6 drives and media.

**Restriction:** If encryption is enabled for a device class, and the device class is associated with a storage pool, the storage pool should not share a scratch pool with other device classes that cannot be encrypted. If a tape is encrypted, and you plan to use it on a drive that cannot be encrypted, you must manually relabel the tape before it can be used on that drive.

**ON** Specifies that Tivoli Storage Manager is the key manager for drive encryption and allows drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes are not encrypted. For example, back up sets, export volumes, and database backup volumes are not encrypted.) If you specify ON and you enable another method of encryption, drive encryption is not allowed and backup operations fail.

**Note:** 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.)

**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is allowed 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 off encryption. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns off encryption.

**OFF**

Specifies that drive encryption is not allowed. If you enable another method of encryption, backups fail. If you enable the application method, Tivoli Storage Manager disables encryption and backups are attempted.

**Example: Define an LTO device class**

Define a device class that is 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)

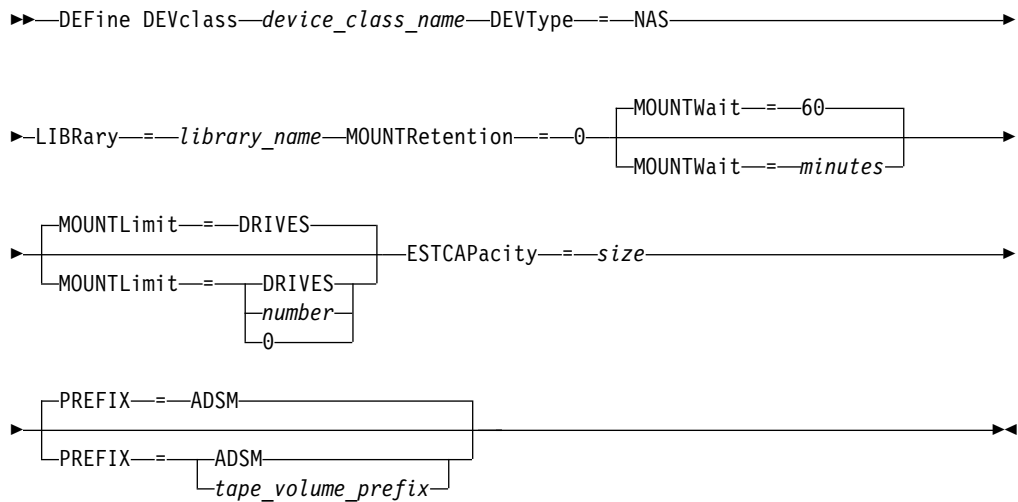
Use the NAS device class when you are using NDMP (Network Data Management Protocol) operations to back up network-attached storage (NAS) file servers. The device class is for drives that are supported by the NAS file server for backups.

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 that are 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 that an idle sequential access volume is retained before it is dismounted. Zero (0) is the only supported value for device classes with DEVTYPE=NAS.

#### MOUNTWAIT

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

#### **ESTCAPacity (Required)**

Specifies the estimated capacity for the volumes that are assigned to this device class.

You must specify this value as an integer followed by one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that is 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
```

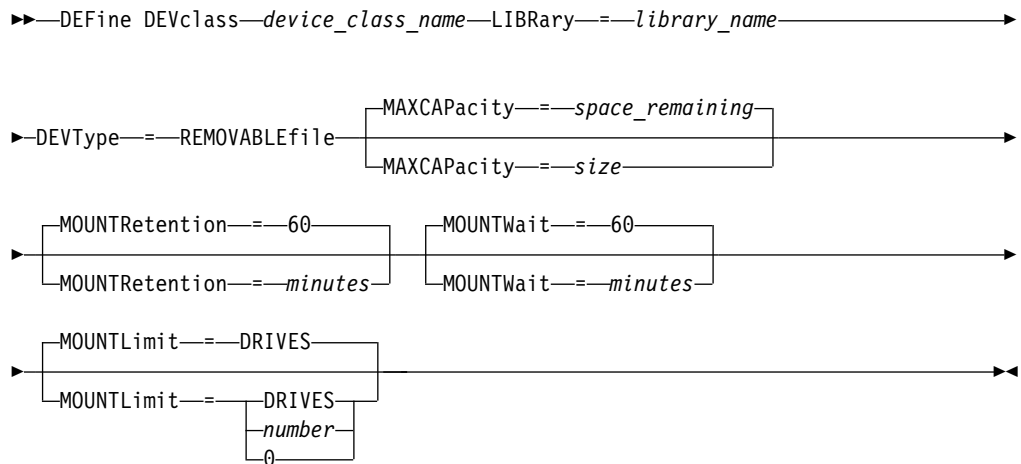
## DEFINE DEVCLASS (Define a REMOVABLEFILE device class)

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.

#### MAXCAPacity

Specifies the maximum size of any volumes that are 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 can 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 that remains on the media after it is first used.

*size*

You must specify this value as an integer followed by K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes).

For example, MAXCAPACITY=5M specifies that the maximum capacity for a volume in this device class is 5 MB. The smallest value that is allowed is 1 MB (that is, MAXCAPACITY=1M).

**MOUNTRetention**

Specifies the number of minutes that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

## DEFINE DEVCLASS (Define a SERVER device class)

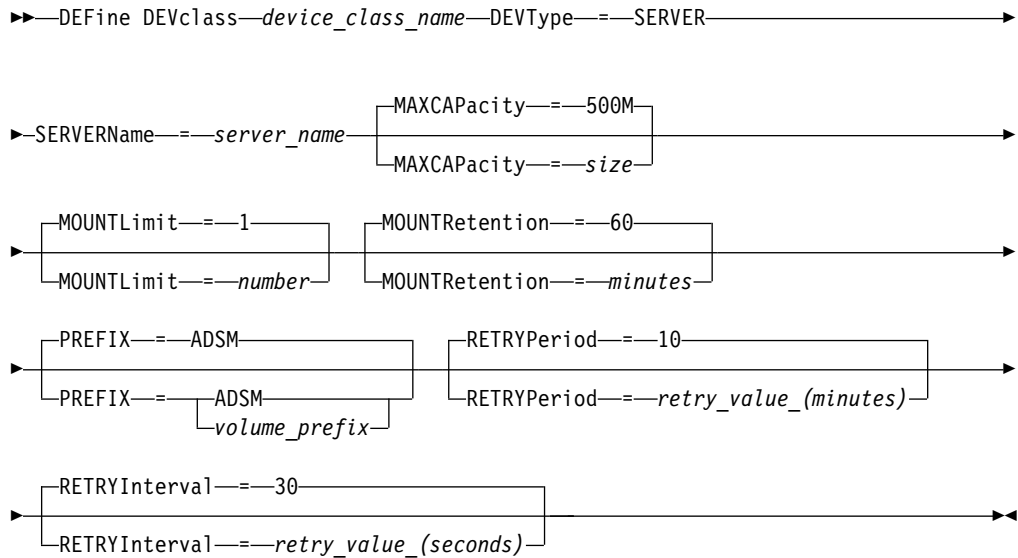
Use the SERVER device class to use storage volumes or files that are archived in another Tivoli Storage Manager server.

If data retention protection is activated with the **SET ARCHIVERETENTIONPROTECTION** command, you cannot define a server 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 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 that are 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), G (gigabytes), or T (terabytes). The minimum value that is allowed is 1 MB (MAXCAPACITY=1M).

**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 1 - 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 the connection closes. This parameter is optional. The default value is 60. You can specify a number 0 - 9999.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**PREFIX**

Specifies the beginning portion of the high-level archive file name on the target server. This parameter is optional. The default is **ADSM**. The maximum length of this prefix is 8 characters.

If you have a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, national, hyphen, or numeric characters.

An example of a high-level archive file name that uses 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. You can specify a number 0 - 9999. The default value is 10 minutes.

**RETRYInterval**

Specifies the retry interval in seconds. The retry interval is how often retries are done within a specific time period. This parameter is optional. You can specify a number 1 - 9999. 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.

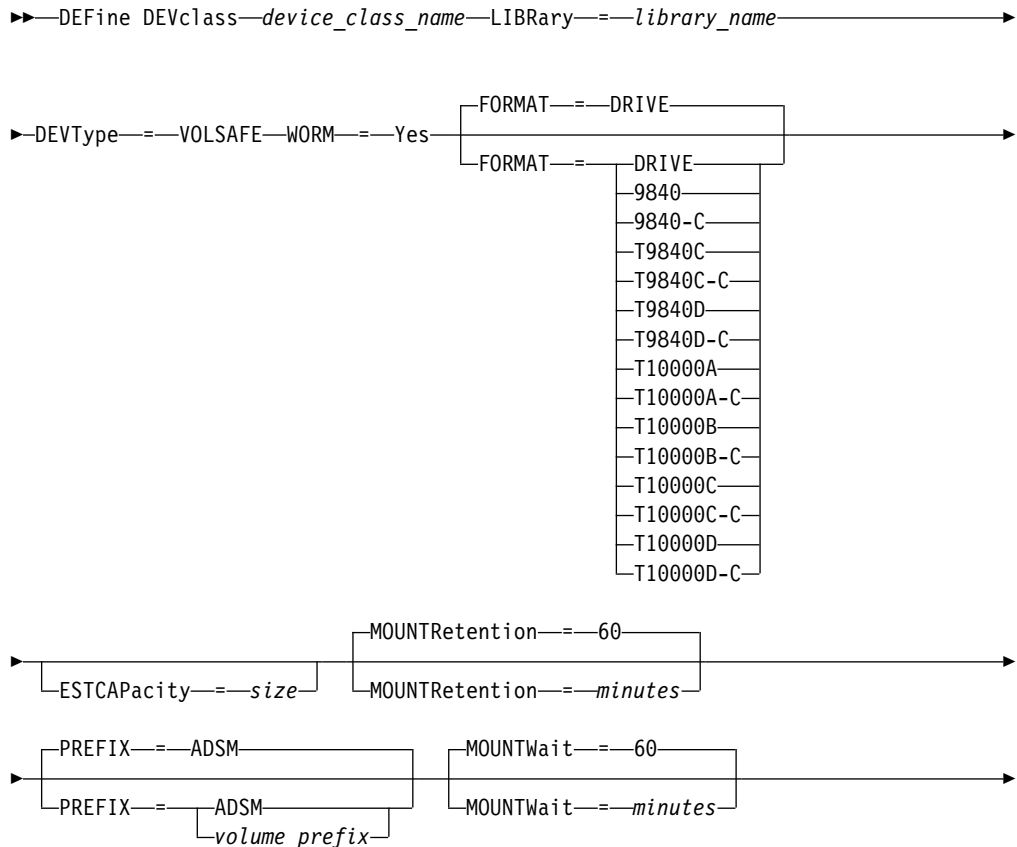
### Restrictions:

1. NAS-attached libraries are not supported.
2. VolSafe media and read/write media must be 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 one time, no additional data can be written to them.

### 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 and T10000 drives.

For information about defining a library object, see “DEFINE LIBRARY (Define a library)” on page 232.

### **DEVType=VOLSAFE** (Required)

Specifies that the VOLSAFE device type is assigned to the device class. The label on this type of cartridge can 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.

### **WORM**

Specifies whether the drives use WORM (write once, read many) media. The parameter is required. The value must be Yes.

#### **Yes**

Specifies that the drives use WORM media.

### **FORMAT**

Specifies the recording format to be used when data is written 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 that is 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 71. 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.  <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 that contains 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
9840-C	See note 80 GB	LZ-1 Enhanced (4:1) compressed format, using an 80 GB cartridge with 270 meters (885 feet) of tape
T9840C	40 GB	Uncompressed T9840C format, using a StorageTek 9840 cartridge
T9840C-C	80 GB	Compressed T9840C format, using a StorageTek 9840 cartridge
T9840D	75 GB	Uncompressed T9840D format, using a StorageTek 9840 cartridge
T9840D-C	150 GB	Compressed T9840D format, using a StorageTek 9840 cartridge
T10000A	500 GB	Uncompressed T10000A format, using a StorageTek T10000 cartridge
T10000A-C	1 TB	Compressed T10000A format, using a StorageTek T10000 cartridge
T10000B	1 TB	Uncompressed T10000B format, using an Oracle StorageTek T10000 cartridge
T10000B-C	2 TB	Compressed T10000B format, using an Oracle StorageTek T10000 cartridge
T10000C	5 TB	Uncompressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge
T10000C-C	10 TB	Compressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge
T10000D	8 TB	Uncompressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge
T10000D-C	15 TB	Compressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge

### ESTCAPacity

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

For more information about the default estimated capacity for cartridge tapes, see Table 71 on page 220.

#### **MOUNTRetention**

Specifies the number of minutes that an idle sequential access volume is retained before it is dismounted. This parameter is optional. The default value is 60 minutes. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **PREFIX**

Specifies the beginning portion of the high-level archive file name on the target server. This parameter is optional. The default is **ADSM**. The maximum length of this prefix is 8 characters.

If you have a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, national, hyphen, or numeric characters.

An example of a high-level archive file name that uses the default prefix is **ADSM.volume1**.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

## DEFINE DOMAIN (Define a new policy domain)

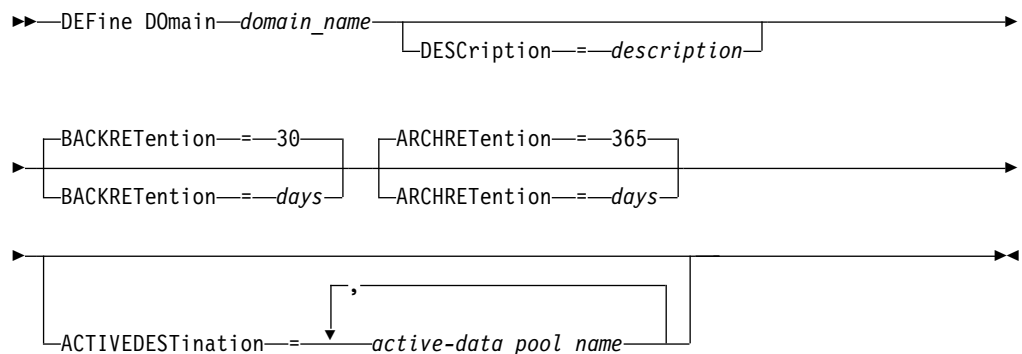
Use this command to define a new policy domain. A policy domain contains policy sets, management classes, and copy groups. A client is assigned to one policy domain. The ACTIVE policy set in the policy domain determines the rules for clients that are assigned to the domain. The rules control the archive, backup, and space management services that are 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 the new management class and the default management class do not contain a backup copy group.
- The management class to which a file is bound no longer exists. 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. 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. 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. The default management class does not contain an archive copy group.

#### **ACTIVEDESTINATION**

This optional parameter specifies the names of active-data pools that store active versions of backup data for nodes that are assigned to the domain. You can specify up to 10 active-data pools for a domain, which is separated by commas. Spaces are not permitted between the names.

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 that is 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 simultaneous-write function is used to write data to an active-data pool, the server verifies that the node meets the criteria during backup operations by Tivoli Storage Manager backup-archive clients or by application clients by using the Tivoli Storage Manager API. The verification is also performed when active-data is being copied by using the **COPY ACTIVE DATA** 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 72. Commands related to DEFINE DOMAIN*

<b>Command</b>	<b>Description</b>
ACTIVATE POLICYSET	Validates and activates a policy set.
COPY DOMAIN	Creates a copy of a policy domain.
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 DOMAIN	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 to make the drive usable by Tivoli Storage Manager. For more information, see “DEFINE PATH (Define a path)” on page 262. If you are using a SCSI or VTL library type, see “PERFORM LIBACTION (Define or delete all drives and paths for a library)” on page 653.

You can define more than one drive for a library by issuing the **DEFINE DRIVE** command for each drive. Stand-alone drives always require a manual library.

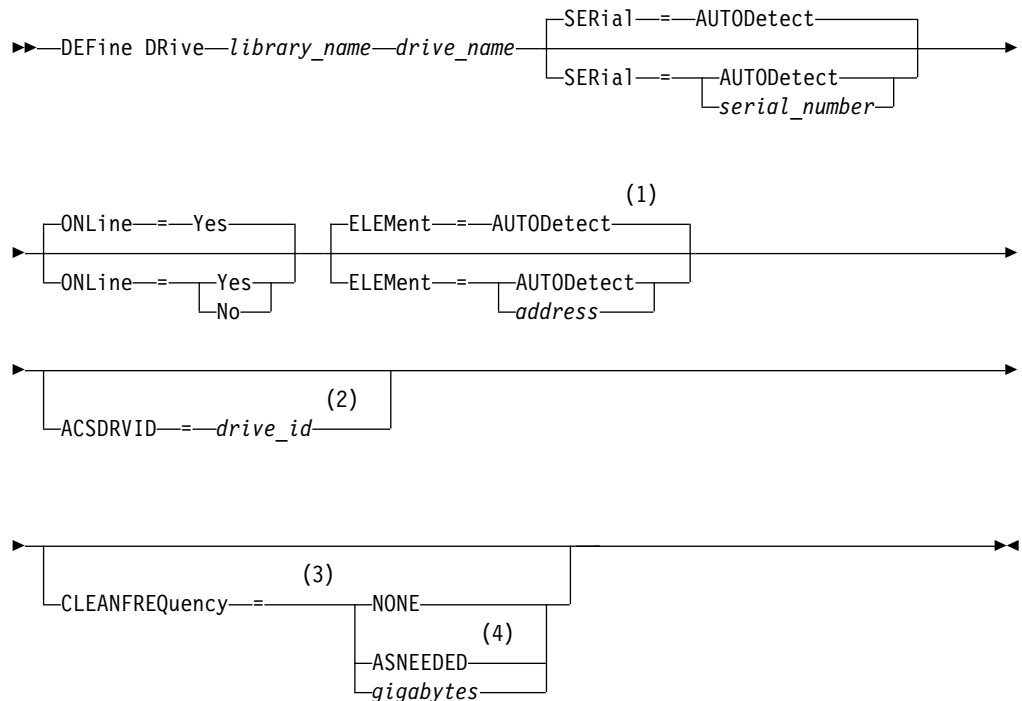
For detailed and current drive support information, see the Supported Devices website for your operating system:

[http://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)

### Privilege class

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

### Syntax



### Notes:

- 1 The **ELEMENT** parameter is only necessary for drives in SCSI libraries when the drive type is a network attached SCSI (NAS) drive.

- 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. For more information, see the parameter description.

## 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 that is 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 is used as the serial number.

If SERIAL=*serial\_number*, then the serial number that is entered is 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 might not be supported. In this case, the serial number is 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 a drive within a SCSI or virtual tape library (VTL). The server uses the element address to connect the physical location of the drive to the SCSI or VTL address of the drive. The default is AUTODETECT.

If ELEMENT=AUTODETECT, then the element number is automatically detected by the server when the path to the drive is defined.

To find the element address for your library configuration, consult the information from the manufacturer.

#### **Restriction:**

- The ELEMENT parameter is valid only for drives in SCSI libraries or VTLs when the drive type is not a network attached SCSI (NAS) drive.
- 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 might not be supported. In this case, you must 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. For the most complete automation of cleaning for an automated library, you must have a cleaner cartridge that is checked into the library's volume inventory.

If you are using library-based cleaning, NONE is advised when your library type supports this function.

This parameter is not valid for externally managed libraries, such as 3494 libraries or StorageTek libraries that are 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.

### 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. See the Supported Devices website for your operating system to view detailed drive information. If **ASNEEDED** is not supported, you can use the *gigabytes* value for automatic cleaning.

For IBM 3592 and LTO drives, library-based cleaning is advised. If library-based cleaning is not supported, then **ASNEEDED** must be used. *Gigabytes* is not recommended.

**Restriction:** Tivoli Storage Manager does not control the drives that are 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.

**Important:** When **CLEANFREQUENCY=gigabyte**, drive cleaning can occur before the gigabyte setting is reached, if the drive notifies the device driver that a cleaning is necessary.

Consult the information from the drive manufacturer 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 that is recommended by IBM for IBM drives ensures that the drives are not overcleaned.

For IBM 3590 drives, 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/rmt/tsmmt0
```

### 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=1,2,3,4
define path server01 acsdrv1 srctype=server desttype=drive
library=acslib device=/dev/rmt/4m
```

### 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/rmt/4m
```

## Related commands

Table 73. 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.
PERFORM LIBACTION	Defines all drives and paths for a library.
QUERY DRIVE	Displays information about drives.
QUERY LIBRARY	Displays information about one or more libraries.

*Table 73. Commands related to DEFINE DRIVE (continued)*

<b>Command</b>	<b>Description</b>
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.

### 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 74. 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

```
»» DEFINE GRPMEMBER group_name member_name
```

### 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 75. 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 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 be accessed by only one source: a Tivoli Storage Manager server or a data mover. However, the drives in a library can be accessed by multiple sources.

The following library types can be defined to the server. Syntax and parameter descriptions are available for each type.

- “DEFINE LIBRARY (Define a 349X library)” on page 234
- “DEFINE LIBRARY (Define an ACSLS library)” on page 237
- “DEFINE LIBRARY (Define an External library)” on page 240
- “DEFINE LIBRARY (Define a FILE library)” on page 242
- “DEFINE LIBRARY (Define a manual library)” on page 243
- “DEFINE LIBRARY (Define a SCSI library)” on page 245
- “DEFINE LIBRARY (Define a shared library)” on page 249
- “DEFINE LIBRARY (Define a VTL library)” on page 250

For detailed and current library support information, see the Supported Devices website for your operating system:

[http://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)

### Related commands

Table 76. Commands related to **DEFINE LIBRARY**

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 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 LIBRARY	Deletes a library.
DELETE PATH	Deletes a path from a source to a destination.
LABEL LIBVOLUME	Labels volumes in manual or automated libraries.
PERFORM LIBACTION	Defines all drives and paths for a library.
QUERY DRIVE	Displays information about drives.
QUERY LIBRARY	Displays information about one or more libraries.
QUERY LIBVOLUME	Displays information about a library volume.

Table 76. Commands related to **DEFINE LIBRARY** (continued)

<b>Command</b>	<b>Description</b>
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 LIBVOLUME	Changes the status of a storage volume.
UPDATE PATH	Changes the attributes associated with a path.

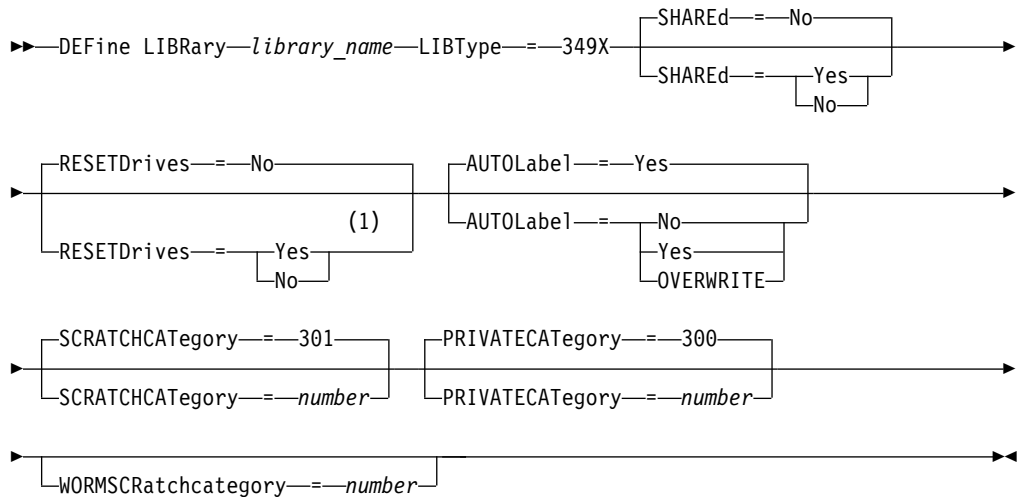
## DEFINE LIBRARY (Define a 349X library)

Use this syntax to define a 349X library.

### Privilege class

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

### Syntax



### Notes:

- 1 The default value of the **RESETDRIVES** parameter is conditional. If the **SHARED** parameter is set to NO, the value of the **RESETDRIVES** parameter is NO. If the **SHARED** parameter is set to YES, the value of the **RESETDRIVES** parameter is YES.

### Parameters

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

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### **LIBType=349X** (Required)

Specifies that the library is an IBM 3494 Tape Library Dataserver or an IBM Tape System Library Manager emulating a 3494 Tape Library Dataserver.

**Restriction:** IBM 3494 libraries support only one unique device type at a time.

#### **SHARED**

Specifies whether this library is shared with other servers in a storage area network (SAN). This parameter is required when you define a library to the library manager.

#### **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.

#### **AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default is YES.

To use this option, you must check in the tapes with **CHECKLABEL=BARCODE** on the **CHECKIN LIBVOLUME** command.

**Restriction:** If you are defining a library that has drives that are attached to a network-attached storage (NAS) device, you must use the **LABEL LIBVOLUME** command to label the volumes for this library.

**No** Specifies that the server does not attempt to label any volumes.

#### **Yes**

Specifies that the server labels only 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.

#### **SCRATCHCategory**

Specifies the category number to be used for scratch volumes in the 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 from the other category numbers in this library.

#### **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 from the other category numbers in this library.

#### **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 from the other category numbers in this library. This parameter is only valid when 3592 WORM volumes are used.

**Restriction:** If the **WORMSCRATCHCATEGORY** is not defined and the **WORM** parameter is set to YES for the device class, the mount operation fails with an error message.

#### **RESETDrives**

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established. If, for example, a storage agent becomes unavailable but is still holding the path to a drive, persistent reserve allows the server to break the storage agent's reservation and access the drive.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg157002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

The following table describes the three possible configurations for drives that are attached to NAS devices.

Table 77. Configurations for drives that are attached to NAS devices..

Library device configuration	The behavior for persistent reserve
The library device is attached to the Tivoli Storage Manager server, and the tape drives are shared by the server and the NAS device.	Drive reservation preemption is supported when the NAS device supports persistent reserve and it is enabled. For more information about setting persistent reserve, see the documentation for your NAS device.
The library device is attached to the Tivoli Storage Manager server and the tape drives are accessed only from the NAS device.	Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation.

**Yes**

Specifies that drive preemption through persistent reserve is used. YES is the default for a library that is defined with SHARED=YES.

**No** Specifies that drive preemption through persistent reserve is not used. NO is the default for a library that is defined with SHARED=NO.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**Example: Define a 3494 library**

Define a library named my3494 with a scratch category number of 550, a private category number of 600, and a WORM scratch category number of 400®

```
define library my3494 libtype=349x scratchcategory=550
privatecategory=600 wormscratchcategory=400
```

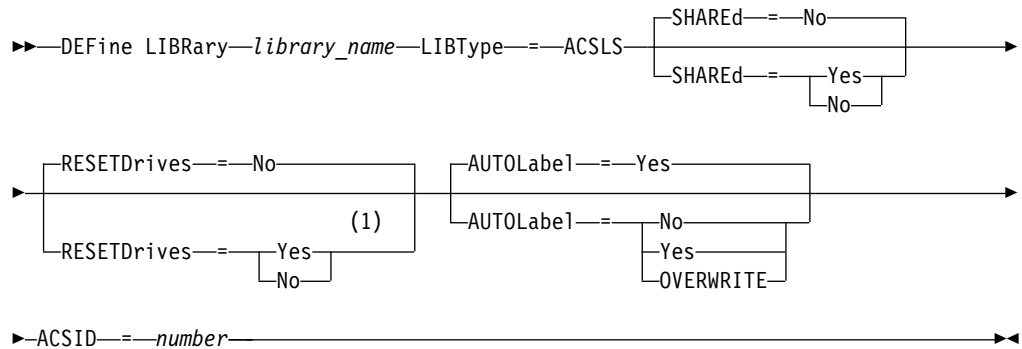
## DEFINE LIBRARY (Define an ACSLS library)

Use this syntax to define an ACSLS library.

### Privilege class

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

### Syntax



### Notes:

- 1 The default value of the **RESETDRIVES** parameter is conditional. If the **SHARED** parameter is set to NO, the value of the **RESETDRIVES** parameter is NO. If the **SHARED** parameter is set to YES, the value of the **RESETDRIVES** parameter is YES.

### Parameters

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

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### **LIBType=ACSLs** (Required)

Specifies that the library is a StorageTek library that is controlled by StorageTek Automated Cartridge System Library Software (ACSLs).

#### **SHARED**

Specifies whether this library is shared with other servers in a storage area network (SAN). This parameter is required when you define a library to the library manager.

#### **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 preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established. If, for example, a storage agent becomes

unavailable but is still holding the path to a drive, persistent reserve allows the server to break the storage agent's reservation and access the drive.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg157002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

The following table describes the three possible configurations for drives that are attached to NAS devices.

Table 78. Configurations for drives that are attached to NAS devices..

Library device configuration	The behavior for persistent reserve
The library device is attached to the Tivoli Storage Manager server, and the tape drives are shared by the server and the NAS device.	Drive reservation preemption is supported when the NAS device supports persistent reserve and it is enabled. For more information about setting persistent reserve, see the documentation for your NAS device.
The library device is attached to the Tivoli Storage Manager server and the tape drives are accessed only from the NAS device.	Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation.

**Yes**

Specifies that drive preemption through persistent reserve is used. YES is the default for a library that is defined with SHARED=YES.

**No** Specifies that drive preemption through persistent reserve is not used. NO is the default for a library that is defined with SHARED=NO.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default is YES.

To use this option, you must check in the tapes with CHECKLABEL=BARCODE on the **CHECKIN LIBVOLUME** command.

**Restriction:** If you are defining a library that has drives that are attached to a network-attached storage (NAS) device, you must use the **LABEL LIBVOLUME** command to label the volumes for this library.

**No** Specifies that the server does not attempt to label any volumes.

**Yes**

Specifies that the server labels only 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.

**ACSID (Required)**

Specifies the number of this StorageTek library that is assigned by the ACSSA (Automatic Cartridge System System Administrator). This number can be from 0 to 126. Issue QUERY ACS on your system to get the number for your library ID. This parameter is required.

For more information, see your StorageTek documentation.

**Example: Define a shared ACSLS library**

Define a library named ACSLIB with the library type of ACSLS and an ACSID of 1.

```
define library acslib libtype=acsls acsid=1 shared=yes
```

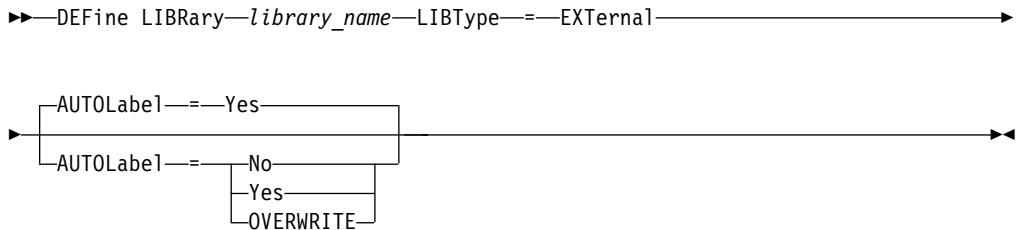
## DEFINE LIBRARY (Define an External library)

Use this syntax to define an External library.

### 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 be defined. The maximum length of this name is 30 characters.

#### **LIBType=EXTERNAL** (Required)

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.

#### **AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default is YES.

To use this option, you must 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 labels only 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.

### 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 you are using Gresham Enterprise DistribuTAPE, the external library manager executable file is in the following directory:

```
/opt/OMIdtelm/bin/elm
```

If you are using the IBM Tape System Library Manager, the external library manager executable file can be found in the following directory:

```
/opt/IBM/TSLM/client/tsm/elm
```

For more information, see the *IBM Tape System Library Manager User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7004001>.

1. Define the library:

```
define library extlib libtype=external
```

2. Define the path:

```
define path server1 extlib srctype=server desttype=library  
externalmanager="/opt/OMIdtelm/bin/elm"
```

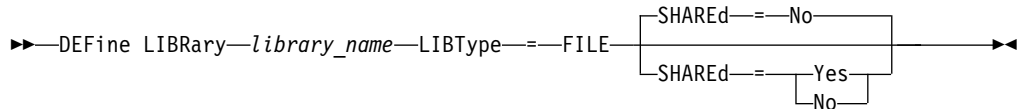
## DEFINE LIBRARY (Define a FILE library)

Use this syntax to define a FILE library.

### 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 be defined. The maximum length of this name is 30 characters.

#### LIBType=FILE (Required)

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. Shared FILE libraries are supported for use in LAN-free backup configurations only. You cannot use a shared FILE library in an environment in which a library manager is used to manage library clients.

#### 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.

#### 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.

### Example: Define a shared FILE library

Define a file library with shared=yes.

```
define library file1 libtype=file shared=yes
```



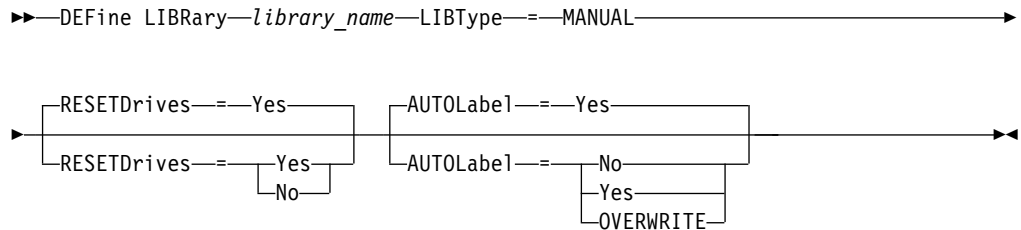
## DEFINE LIBRARY (Define a manual library)

Use this syntax to define a manual library.

### 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 be defined. The maximum length of this name is 30 characters.

#### `LIBType=MANUAL` (Required)

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.

#### `AUTOLabel`

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default is YES.

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.

#### `RESETDrives`

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established. If, for example, a storage agent becomes unavailable but is still holding the path to a drive, persistent reserve allows the server to break the storage agent's reservation and access the drive.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

**Yes**

Specifies that drive preemption through persistent reserve is used. YES is the default for a library that is defined with SHARED=YES.

**No** Specifies that drive preemption through persistent reserve is not used. NO is the default for a library that is defined with SHARED=NO.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**Example: Define a manual library**

Define a library named MANUALMOUNT with the library type of MANUAL.

```
define library manualmount libtype=manual
```

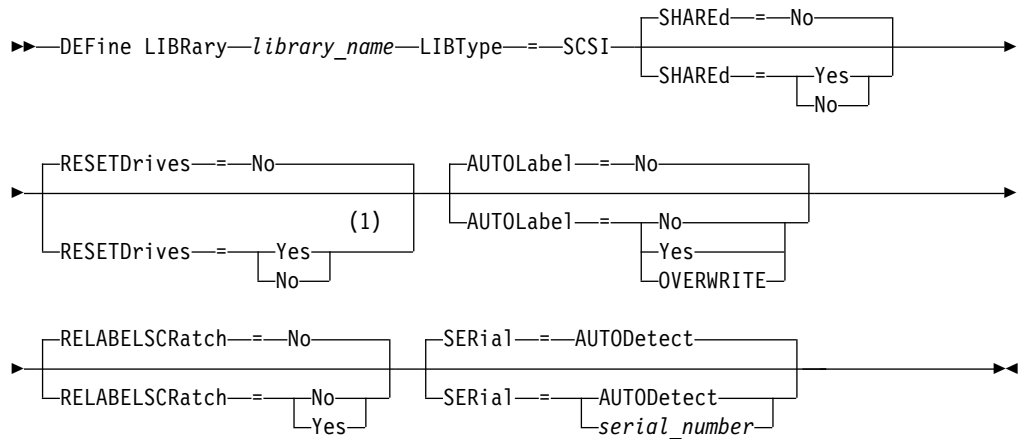
## DEFINE LIBRARY (Define a SCSI library)

Use this syntax to define a SCSI library.

### Privilege class

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

### Syntax



### Notes:

- 1 The default value of the **RESETDRIVES** parameter is conditional. If the **SHARED** parameter is set to NO, the value of the **RESETDRIVES** parameter is NO. If the **SHARED** parameter is set to YES, the value of the **RESETDRIVES** parameter is YES.

### Parameters

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

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### LIBType=SCSI (Required)

Specifies that the library has a SCSI-controlled media changer device. To mount volumes on drives in this type of library, the server uses the media changer device.

#### SHARED

Specifies whether this library is shared with other servers in a storage area network (SAN). This parameter is required when you define a library to the library manager.

#### 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.

### **AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default is NO.

To use this option, you must check in the tapes with CHECKLABEL=BARCODE on the **CHECKIN LIBVOLUME** command.

**Restriction:** If you are defining a library that has drives that are attached to a network-attached storage (NAS) device, you must use the **LABEL LIBVOLUME** command to label the volumes for this library.

**No** Specifies that the server does not attempt to label any volumes.

#### **Yes**

Specifies that the server labels only 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 were 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).

If you have both virtual and real volumes in your VTL, both types are relabeled when this parameter is enabled. If the VTL includes real volumes, specifying this option might impact performance.

**Restriction:** If you are defining a library that has drives that are attached to a network-attached storage (NAS) device, you must use the **LABEL LIBVOLUME** command to label the volumes for this library.

**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.

### **RESETDrives**

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established. If, for example, a storage agent becomes unavailable but is still holding the path to a drive, persistent reserve allows the server to break the storage agent's reservation and access the drive.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.

- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

The following table describes the three possible configurations for drives that are attached to NAS devices.

Table 79. Configurations for drives that are attached to NAS devices

Library device configuration	The behavior for persistent reserve
The library device is attached to the Tivoli Storage Manager server, and the tape drives are shared by the server and the NAS device.	Drive reservation preemption is supported when the NAS device supports persistent reserve and it is enabled. For more information about setting persistent reserve, see the documentation for your NAS device.
The library device is attached to the Tivoli Storage Manager server and the tape drives are accessed only from the NAS device.	Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation.
The library device is attached to the NAS device and accessed indirectly by NDMP (network data management protocol), and the tape drives are accessed only from the NAS device.	Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation.

**Yes**

Specifies that drive preemption through persistent reserve is used. YES is the default for a library that is defined with SHARED=YES.

**No** Specifies that drive preemption through persistent reserve is not used. NO is the default for a library that is defined with SHARED=NO.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**SERial**

Specifies the serial number for the library that is 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 is used as the serial number.

If SERIAL=*serial\_number*, then the number you entered is compared to the number detected by the server.

**Attention:** Depending on the capabilities of the device, SERIAL=AUTODETECT might not be supported. In this case, the serial number is reported as blank.

**Example: Define a SCSI library**

Define a library that is named SCsilIB with a library type of SCSI.

```
define library scsilib libtype=scsi
```

The library requires a path. The device name for the library is:

```
/dev/rmt/1b0
```

Define the path:

```
define path server1 scsilib srctype=server desttype=library  
    device=/dev/rmt/1b0
```

## DEFINE LIBRARY (Define a shared library)

Use this syntax to define a shared library.

### Privilege class

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

### Syntax

```
►►—DEFine LIBRary—library_name—LIBType—==—SHARed—————►  
►—PRIMarylibmanager—==—server_name—————►◄
```

### Parameters

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

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### LIBType=SHARed (Required)

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.

**Important:** Specify this library type when you define the library on a library client.

#### 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.

### Example: Define a shared library

In a SAN, define a library named SHAREDTSM to a library client server named LIBMGR1

```
define library sharedtsm libtype=shared primarylibmanager=libmgr1
```

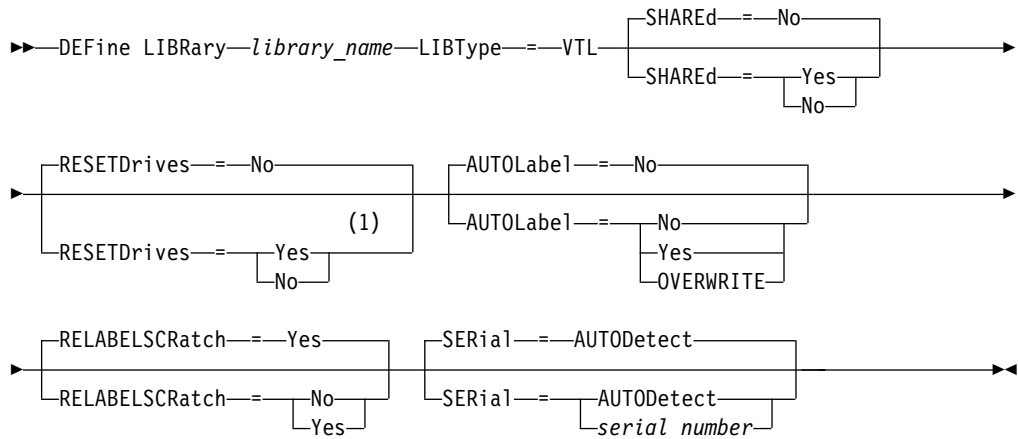
## DEFINE LIBRARY (Define a VTL library)

Use this syntax to define a library that has a SCSI-controlled media changer device that is represented by a virtual tape library (VTL).

### Privilege class

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

### Syntax



### Notes:

- 1 The default value of the **RESETDRIVES** parameter is conditional. If the **SHARED** parameter is set to NO, the value of the **RESETDRIVES** parameter is NO. If the **SHARED** parameter is set to YES, the value of the **RESETDRIVES** parameter is YES.

### Parameters

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

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### LIBType=VTL (Required)

Specifies that the library has a SCSI-controlled media changer device that is represented by a virtual tape library. To mount volumes in drives in this type of library, the server uses the media changer device.

If you are defining a VTL library, your environment must not include any mixed-media and paths must be defined between all drives in the library and all defined servers, including storage agents, that use the library. If either of these characteristics are not true, the overall performance can degrade to the same levels as the SCSI library type; especially during times of high stress.

#### SHARED

Specifies whether this library is shared with other servers in a storage area network (SAN). This parameter is required when you define a library to the library manager.

#### 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 preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established. If, for example, a storage agent becomes unavailable but is still holding the path to a drive, persistent reserve allows the server to break the storage agent's reservation and access the drive.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

#### **Yes**

Specifies that drive preemption through persistent reserve is used. YES is the default for a library that is defined with SHARED=YES.

**No** Specifies that drive preemption through persistent reserve is not used. NO is the default for a library that is defined with SHARED=NO.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

#### **AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional. The default is NO.

To use this option, you must check in the tapes with CHECKLABEL=BARCODE on the **CHECKIN LIBVOLUME** command.

**Restriction:** If you are defining a library that has drives that are attached to a network-attached storage (NAS) device, you must use the **LABEL LIBVOLUME** command to label the volumes for this library.

**No** Specifies that the server does not attempt to label any volumes.

#### **Yes**

Specifies that the server labels only 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 were 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.

If you have both virtual and real volumes in your VTL, both types are relabeled when this parameter is enabled. If the VTL includes real volumes, specifying this option might impact performance.

**Restriction:** If you are defining a library that has drives that are attached to a network-attached storage (NAS) device, you must use the **LABEL LIBVOLUME** command to label the volumes for this library.

#### **Yes**

Specifies that the server relabels volumes that are deleted and returned to scratch. YES is the default.

**No** Specifies that the server does not relabel volumes that are deleted and returned to scratch.

### **SERial**

Specifies the serial number for the library that is 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 is used as the serial number.

If SERIAL=*serial\_number*, then the number you entered is compared to the number detected by the server.

**Attention:** Depending on the capabilities of the device, SERIAL=AUTODETECT might not be supported. In this case, the serial number is reported as blank.

### **Example: Define a VTL library**

Define a library named VTL LIB with a library type of VTL.

```
define library vtllib libtype=vtl
```

The library requires a path. The device name for the library is:

```
/dev/rmt/1b0
```

Define the path:

```
define path server1 vtllib srctype=server desttype=library  
device=/dev/rmt/1b0
```

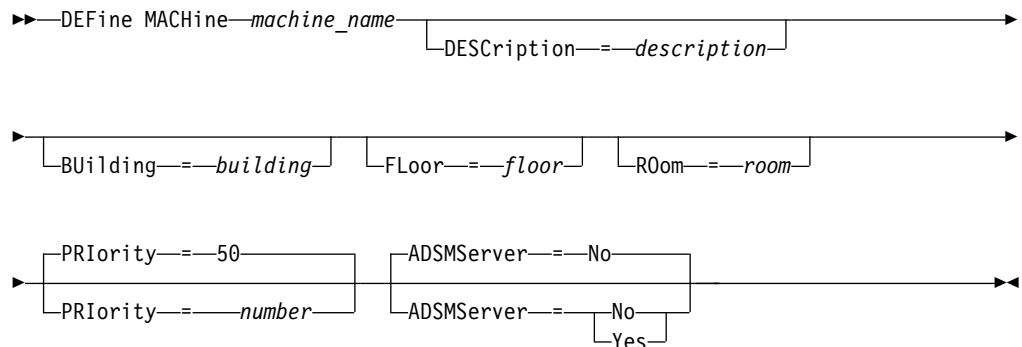
## DEFINE MACHINE (Define machine information for disaster recovery)

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.

#### ADSM\_SERVER

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:

**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 80. Commands related to DEFINE MACHINE*

<b>Command</b>	<b>Description</b>
DEFINE MACHNODEASSOCIATION	Associates a 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 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.

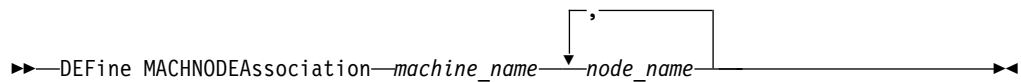
A node remains associated with a machine unless the node, the machine, or the association itself is deleted.

### 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 81. 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 node 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.



## 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.

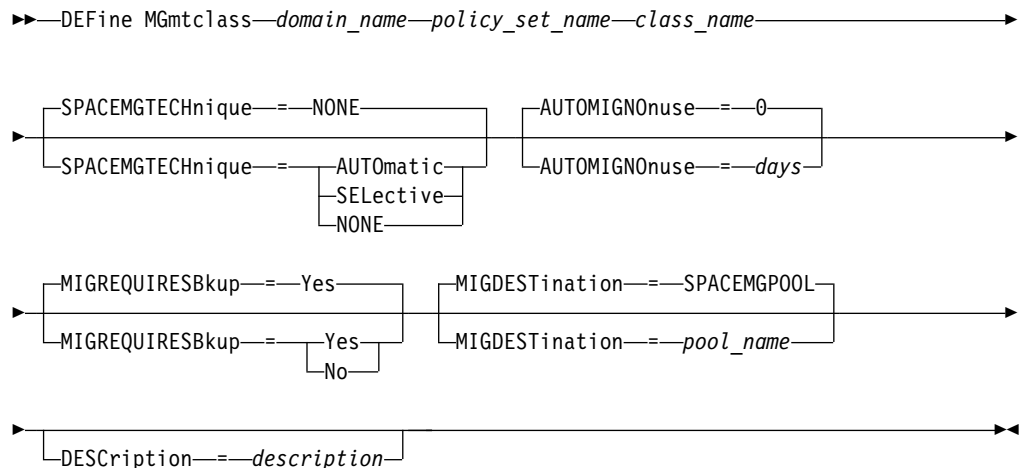
You can define one or more management classes for each policy set in a policy domain. A management class can contain a backup copy group, an archive copy group, or both. The user of a client node can select any management class in the active policy set or use the default management class.

**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 that is using this management class is eligible for migration. This parameter is optional. The default is NONE. 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 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 in the range 0 - 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 that are 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.

Your choice for the destination might depend on factors such as the following:

- The number of client nodes that are migrated to the storage pool. When many user files are stored in the same storage pool, volume contention can occur as users try to migrate files to or recall files from the storage pool.
- How quickly the files must be recalled. If you need immediate access to migrated versions, you can specify a disk storage pool as the destination.

The command fails if you specify a copy storage pool or an active-data 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 that is 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 SMPOOL 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 82. 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 64 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 83. 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

```
►► DEFINE NODEGROUPMEMBER group_name node_name ◀◀
```

### 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 84. 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 for a source to access a destination. Both the source and destination must be defined 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 the server.

Syntax and parameter descriptions are available for the following path types.

- “DEFINE PATH (Define a path when the destination is a drive)” on page 263
- “DEFINE PATH (Define a path when the destination is a library)” on page 268

For detailed and current device support information, see the Supported Devices website for your operating system:

[http://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)

### Related commands

Table 85. Commands related to **DEFINE PATH**

Command	Description
DEFINE DATAMOVER	Defines a data mover to the 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.
PERFORM LIBACTION	Defines all drives and paths for a library.
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.

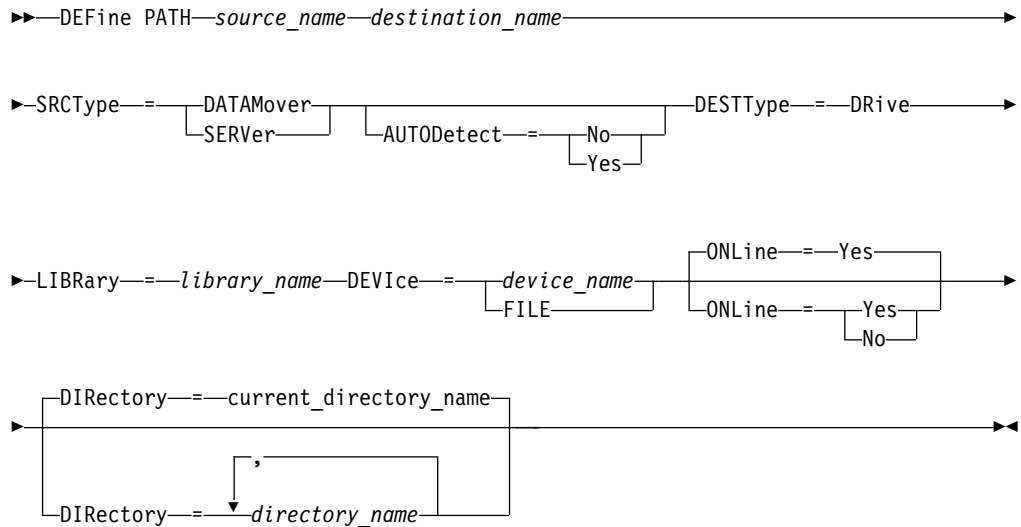
## DEFINE PATH (Define a path when the destination is a drive)

Use this syntax when you define a path to a drive.

### Privilege class

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

### Syntax



### 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.

#### **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 is automatically updated in the database at the time that the path is defined. This parameter is optional. This parameter is only valid for paths that are defined from the local server to a drive. Possible values are:

**No** Specifies that the serial number is not 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 is not automatically updated to reflect the same serial number that the drive reports to the server.

**Important:**

1. If you did not set the serial number when you defined the drive, the server always tries to detect the serial number, and AUTODETECT defaults to YES. If you 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 definition is updated with the detected serial number.
3. If you set DESTTYPE=DRIVE and AUTODETECT=YES, then the drive element number in the database is 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 might not be supported.

**DESTType=DRive (Required)**

Specifies that a drive is the destination. When the destination is a drive, you must specify a library name.

**LIBRARY**

Specifies the name of the library to which the drive is assigned. The library and its drives must already be defined to the 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. See Table 86 for examples.

Table 86. Examples of device names

Source to destination	Example
Server to a drive (not a FILE drive)	/dev/rmt/tsmmt3
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 drive	NetApp NAS file server: rst01 EMC Celerra NAS file server: c436t011 IBM System Storage N Series: rst01

**Important:**

- For information about the device name when the source is a storage agent, see the Tivoli Storage Manager for Storage Area Networks product information.
- For 349X libraries, the alias name is a symbolic name that is specified in the /etc/ibmat1.conf file. For more information, see *IBM Tape Device Drivers Installation and User's Guide*, which can be downloaded from the IBM Systems support site at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972>.

- 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
```

### **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.

### **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.

For a path from a storage agent to a FILE device, 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 that is 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 runtime failure.

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 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, it experiences 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 that are specified with this parameter are not validated on the server.
2. Tivoli Storage Manager does not create shares or permissions, or mount the target file system. You must complete these actions before you start the storage agent.

**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 that is shared with the 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"
```

**Example: Configure a storage agent to use a FILE library**

The following example 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. 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 stal class1 srctype=server desttype=drive device=file
directory="/opt/ibm1,/opt/ibm2,/opt/ibm3" library=classa
```

In this scenario, the storage agent, STA1, replaces 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 is still able to access file volume /opt/tivoli1/file1.dsm, but the storage agent STA1 is not 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 that is associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume is still accessible from the 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.

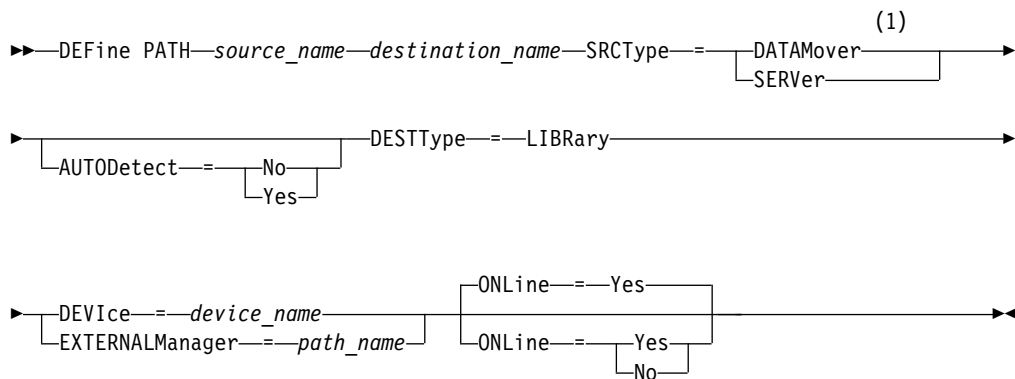
## DEFINE PATH (Define a path when the destination is a library)

Use this syntax when defining a path to a library.

### Privilege class

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

### Syntax



### 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.

**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 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. Depending on the capabilities of the device, the AUTODETECT parameter may not be supported.

**DESTType=LIBRARY (Required)**

Specifies that a library is the destination. This parameter is required.

**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 library. See Table 87 for examples.

Table 87. Examples of device names

Source to destination	Example
Server to a library	/dev/rmt/tsm1b4
Storage agent to a drive when the drive is a logical drive in a FILE library	FILE
NAS data mover to a library	mc0

**Important:**

- For information about the device name when the source is a storage agent, see the Tivoli Storage Manager for Storage Area Networks product information.
- For 349X libraries, the alias name is a symbolic name that is specified in the /etc/ibmatl.conf file. For more information, see *IBM Tape Device Drivers Installation and User's Guide*, which can be downloaded from the IBM Systems support site at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972>.
- 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 Tivoli Storage Manager can send media access requests. Use single quotation marks around the value of this parameter. For example, enter:

```
/opt/GESedt-acsls/bin/elmdt
```

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.

**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.

**Example: Define a path from a server to a library**

Define a path from the server SATURN to the SCSI type library SCSILIB:

```
define path saturn scsilib srctype=server  
desttype=library device=/dev/rmt/tsmlb3
```



Table 88. Commands related to **DEFINE POLICYSET** (continued)

<b>Command</b>	<b>Description</b>
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:

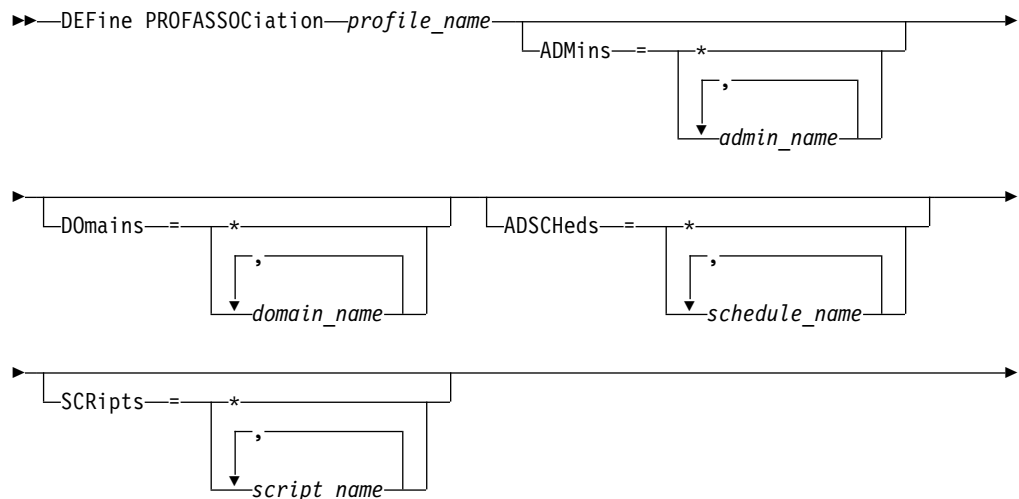
- Administrator registrations and authorities
- Policy domains, which include the domains' policy sets, management classes, copy groups, and client schedules
- Administrative schedules
- Server command scripts
- Client option sets
- Server definitions
- Server group definitions

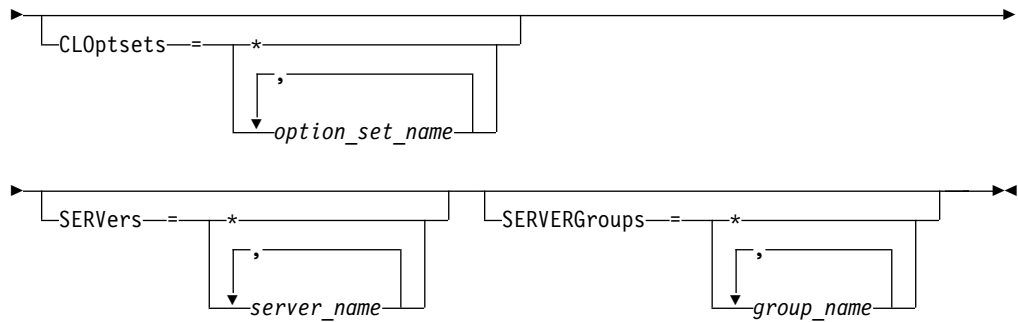
**Tip:** The configuration manager does not distribute status information for an object to managed servers. For example, information such as the number of days since an administrator last accessed the server is not distributed to managed servers. This type of information is maintained in the databases 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.

### D0mains

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.



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.

### **Important:**

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

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 89. Commands related to DEFINE PROFASSOCIATION*

<b>Command</b>	<b>Description</b>
<b>COPY PROFILE</b>	Creates a copy of a profile.
<b>DEFINE PROFILE</b>	Defines a profile for distributing information to managed servers.
<b>DELETE PROFASSOCIATION</b>	Deletes the association of an object with a profile.
<b>DELETE PROFILE</b>	Deletes a profile from a configuration manager.
<b>LOCK PROFILE</b>	Prevents distribution of a configuration profile.

Table 89. Commands related to DEFINE PROFASSOCIATION (continued)

<b>Command</b>	<b>Description</b>
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 90. 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 90. Commands related to DEFINE PROFILE (continued)*

<b>Command</b>	<b>Description</b>
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

```
►►—DEfIne RECMEDMACHAssociation—media_name—machine_name—►►
```

### 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 91. 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 91. Commands related to DEFINE RECMEDMACHASSOCIATION (continued)*

<b>Command</b>	<b>Description</b>
QUERY RECOVERYMEDIA	Displays media available for machine recovery.



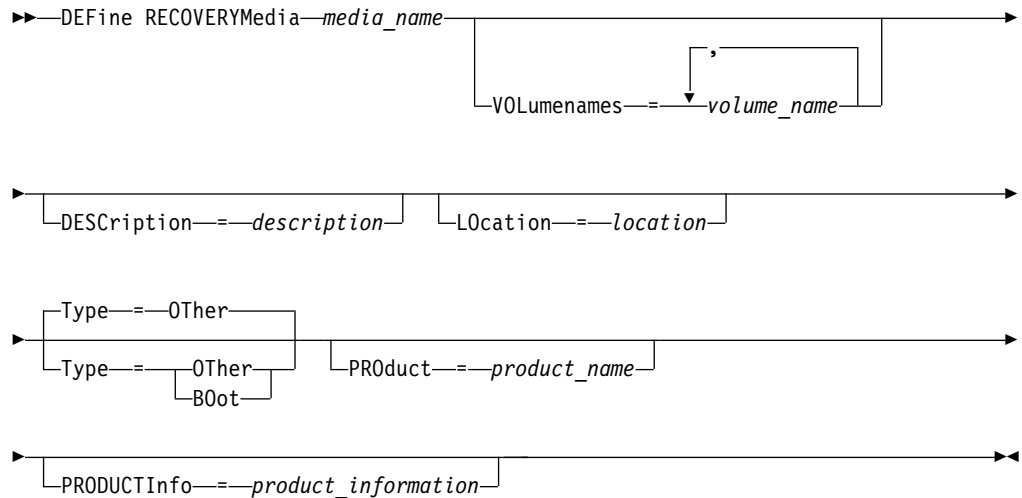
## DEFINE RECOVERYMEDIA (Define recovery media)

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.

#### **L0cation**

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 92. 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.

- “DEFINE SCHEDULE (Define a schedule for an administrative command)” on page 300
- “DEFINE SCHEDULE (Define a client schedule)” on page 286

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 93. 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 MAXSCHEDESESSIONS	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 schedule)

Use the **DEFINE SCHEDULE** command to define a client schedule. Tivoli Storage Manager uses this schedule to automatically perform a variety of client 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.

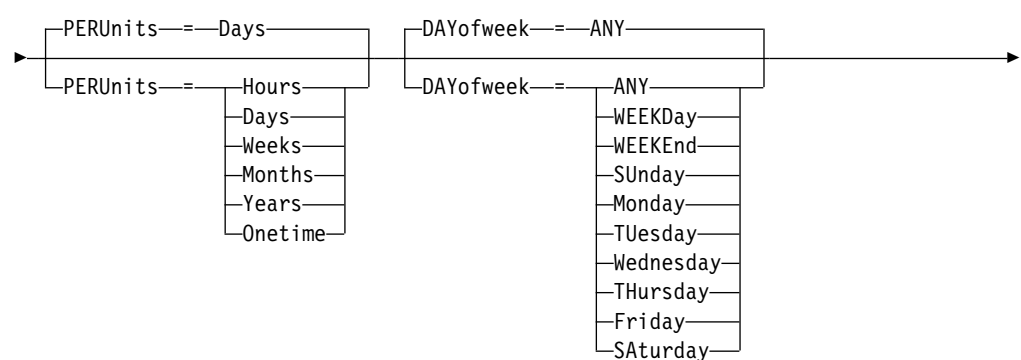
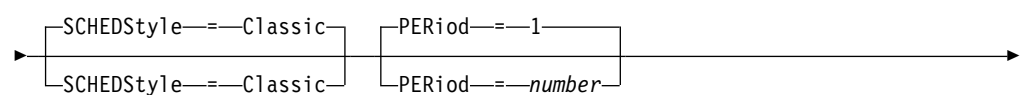
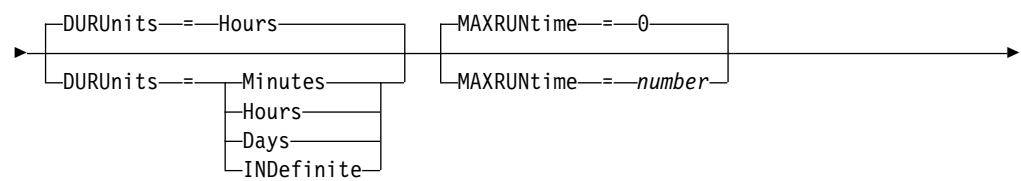
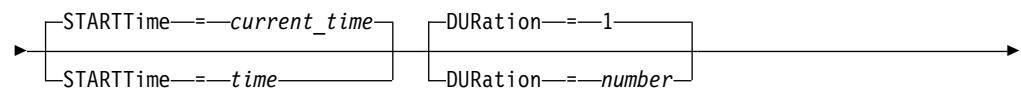
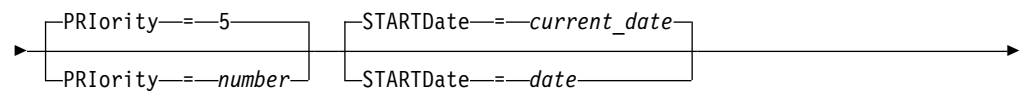
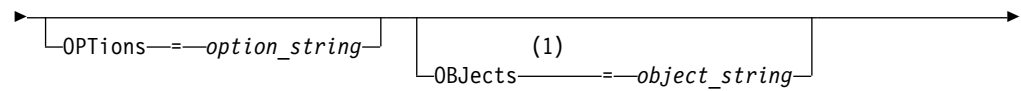
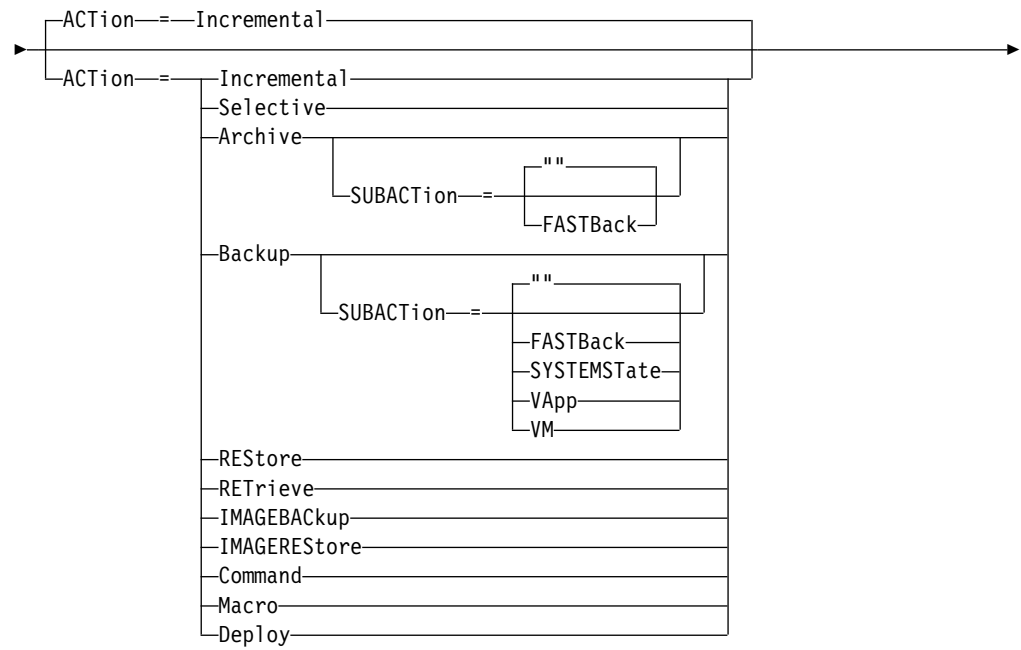
### Syntax

#### Classic client schedule

```
▶▶—DEfine SChedule—domain_name—schedule_name—Type—=—Client—▶
```

```
▶ DESCRiption—=—description—▶
```



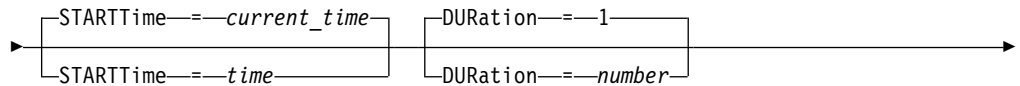
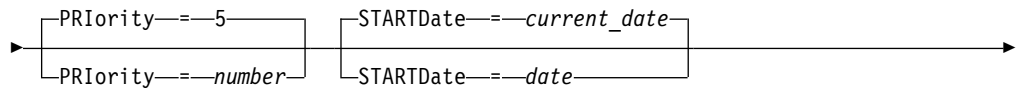
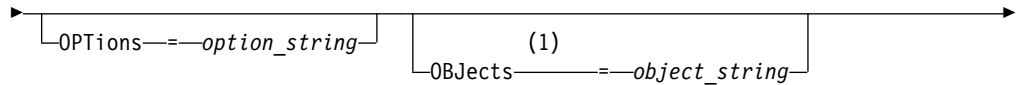
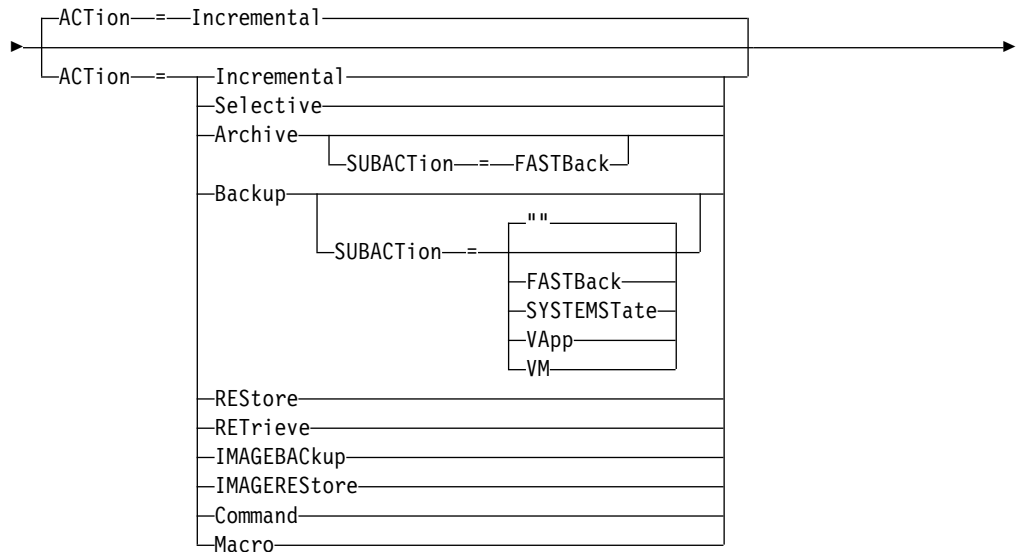
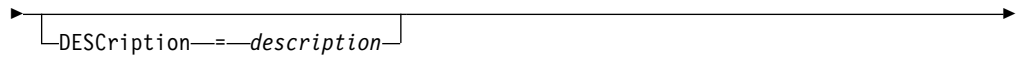
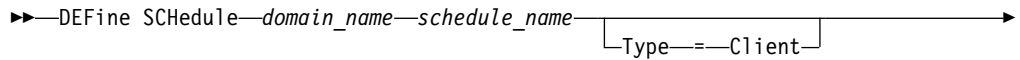


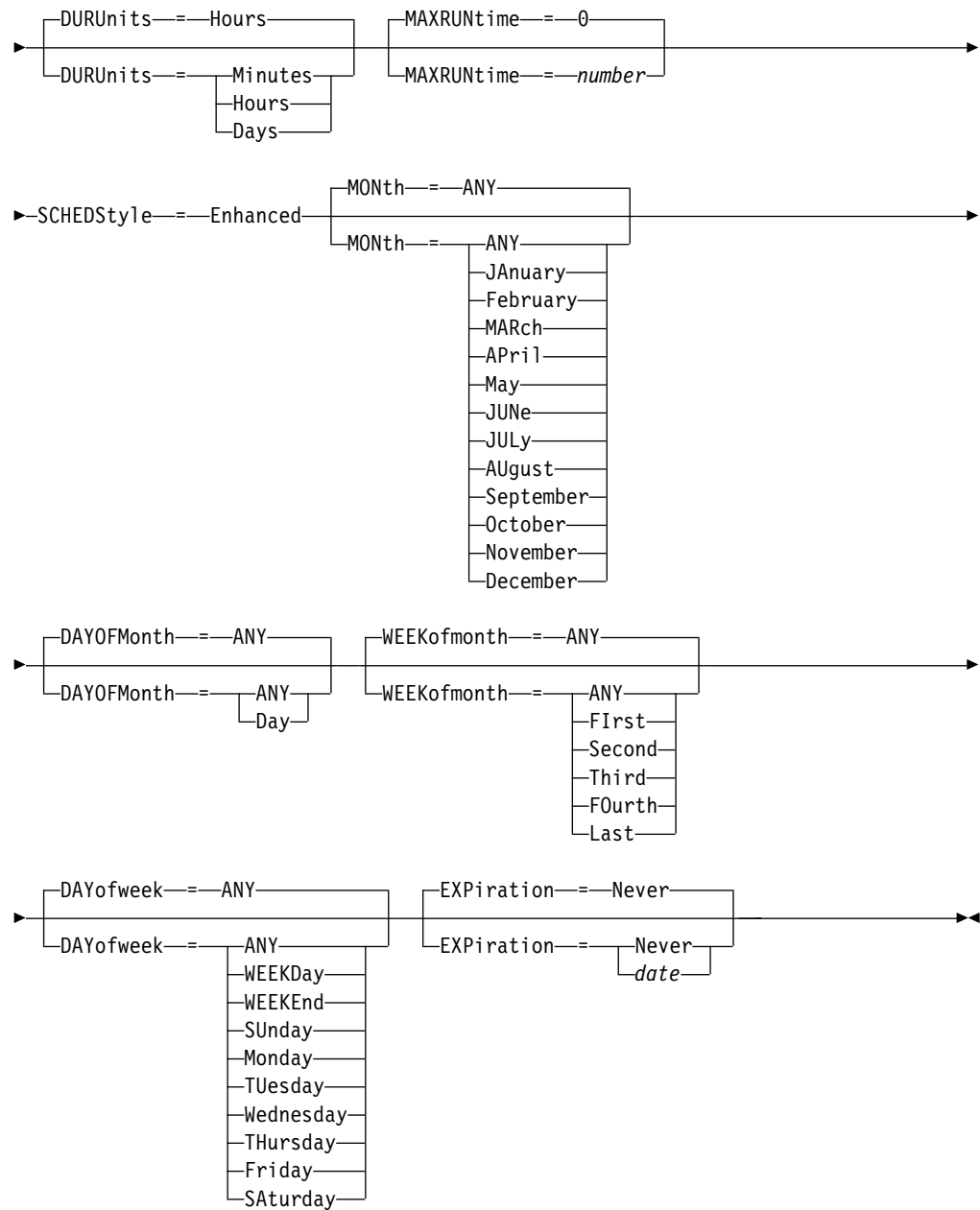
**Notes:**

- 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.

**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 might 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.

**Backup**

Specifies that the schedule backs up 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**

Indicates that the schedule retrieves files that are specified with the OBJECTS parameter.

**Remember:** A second file that is specified acts as the retrieve destination. If you need to retrieve multiple groups of files, create a separate schedule for each group of files.

**IMAGEBACKup**

Specifies that the schedule backs up logical volumes that are specified with the OBJECTS parameter.

**IMAGERESore**

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.

**SUBACTION**



You can specify one of the following values:

"" When a null string (two double quotes) is specified with **ACTION=BACKUP** the backup is an incremental.

#### **FASTBack**

Specifies that a FastBack client operation that is identified by the **ACTION** parameter is to be scheduled for processing. The **ACTION** parameter must be either **ARCHIVE** or **BACKUP**.

#### **SYSTEMState**

Specifies that a client Systemstate backup is scheduled.

#### **VApp**

Specifies that a client vApp backup is scheduled. A vApp is a collection of pre-deployed virtual machines.

**VM** Specifies that a client VMware backup operation is scheduled.

#### **Deploy**

Specifies whether to update client workstations with deployment packages that are specified with the **OBJECTS** parameter. The **OBJECTS** parameter must contain two specifications, the package files to retrieve and the location from which to retrieve them. Ensure that the objects are in the order *files location*. For example:

```
define schedule standard deploy_1 action=DEPLOY objects=  
"\\IBM_ANR_WIN\c$\tsm\maintenance\client\v6r2\Windows\X32\v620\v6200\*  
..\IBM_ANR_WIN\"
```

Values for the following options are restricted when you specify **ACTION=DEPLOY**:

#### **PERUNITS**

Specify **PERUNITS=ONETIME**. If you specify **PERUNITS=PERIOD**, the parameter is ignored.

#### **DURUNITS**

Specify **MINUTES**, **HOURS**, or **DAYS** for the **DURUNITS** parameter. Do not specify **INDEFINITE**.

#### **SCHEDSTYLE**

Specify the default style, **CLASSIC**.

The **SCHEDULE** command fails if the parameters do not conform to the required parameter values, such as the V.R.M.F.

#### **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 can 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 result in an error or are ignored when running the schedule from the server. For example, do not include the following options because they have no impact 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 apostrophes. Enclose individual options that contain spaces in quotation marks. 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. 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 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 `/opt/file 2`, `/opt/gif files`, and `/opt/my test file`, enter:  
`OBJECTS='"/opt/file 2" "/opt/gif files" "/opt/my test file"'`
- To specify `/opt/test file`, enter:  
`OBJECTS='"/opt/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
<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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

**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

Value	Description	Example
<b>NOW+HH:MM</b> or <b>+HH:MM</b>	The current time plus hours and minutes specified	NOW+02:00 or +02:00.  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 <b>-HH:MM</b>	The current time minus hours and minutes specified	NOW-02:00 or -02:00.  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. You can specify one of the following values:

#### **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.

### **MAXRUNTime**

Specifies the maximum run time, which is the number of minutes during which all client sessions that are started by the scheduled operation should be completed. If sessions are still running after the maximum run time, the server issues a warning message, but the sessions continue to run.

**Tip:** The maximum run time is calculated from the beginning of the startup window and not from the time that sessions start within the startup window.

#### **Restrictions:**

- The value of the parameter is not distributed to servers that are managed by an enterprise configuration manager.
- The value of the parameter is not exported by the **EXPORT** command.

The parameter is optional. You can specify a number in the range 0-1440. The default value is 0. A value of 0 means that the maximum run time is indefinite, and no warning message is issued. The maximum run time must be greater than the startup window duration, which is defined by the **DURATION** and **DURUNITS** parameters.

For example, if the start time of a scheduled operation is 9:00 PM, and the duration of the startup window is 2 hours, the startup window is 9:00 PM - 11:00 PM. If the maximum run time is 240 minutes, that is, 4 hours, all client sessions for this operation should be completed by 1:00 AM. If one or more sessions are still running after 1:00 AM, the server issues a warning message.

**Tip:** Alternatively, you can specify a *Run time alert* value of 1:00 AM in the Tivoli Storage Manager Operations Center.

### **SCHEDStyle**

This parameter is optional. `SCHEDSTYLE` defines either the interval between times when a schedule can run, or the days on which it runs. 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. You can specify one of the following values:

#### **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 was 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, which means that the schedule runs 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 runs on each of the specified days of the month. If multiple values resolve to the same day, the schedule runs only once that day.

The default value is ANY. ANY means that the schedule runs 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 runs during each of the specified weeks of the month. If multiple values resolve to the same week, the schedule runs only once during that week.

The default value is ANY. ANY means that the schedule runs 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. You can specify one of the following values:

##### **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 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:

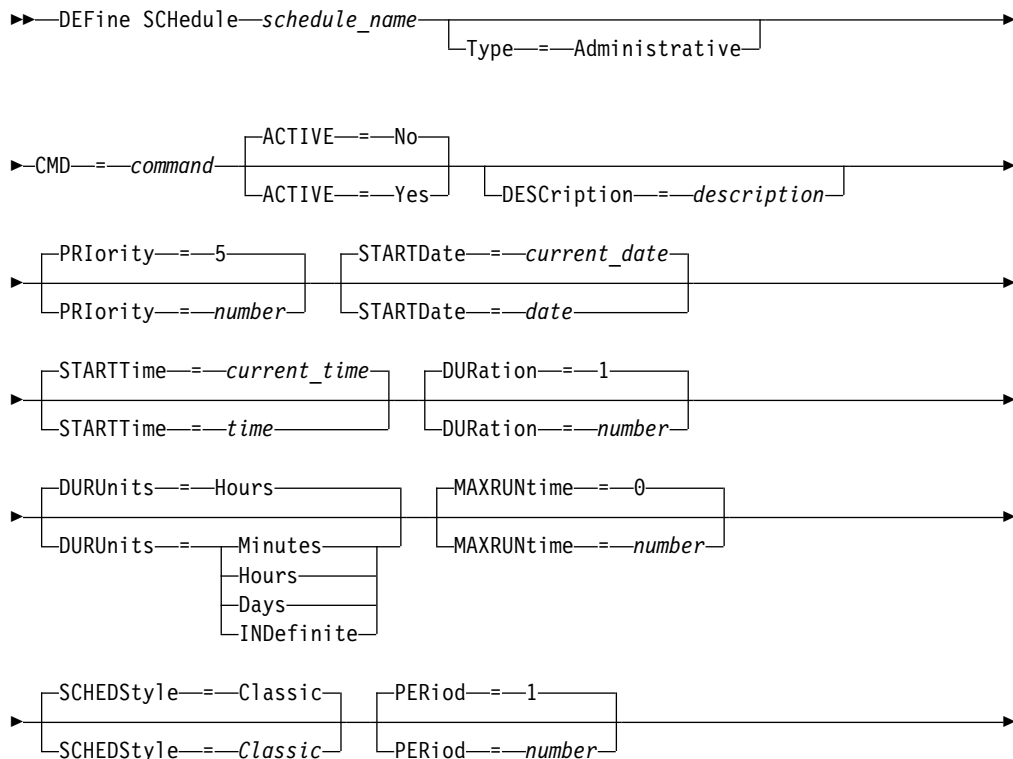
1. You cannot schedule the **MACRO** command or the **QUERY ACTLOG** command.
2. If you are scheduling a command that specifies the **WAIT** parameter, the parameter must be set to **YES** in order for the process to provide a return code to the session that started it. For more information about the **WAIT** parameter, see "Server command processing" on page 16.

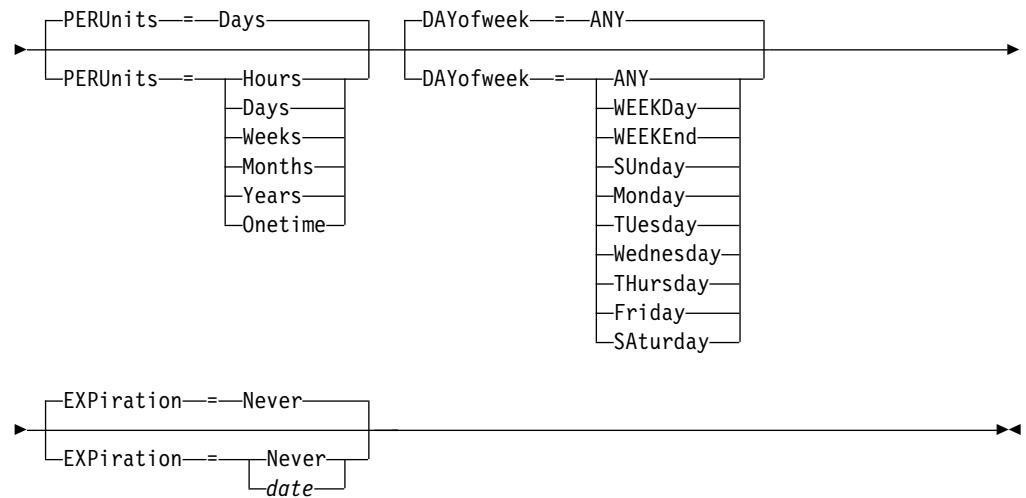
### Privilege class

To define an administrative command schedule, you must have system privilege.

### Syntax

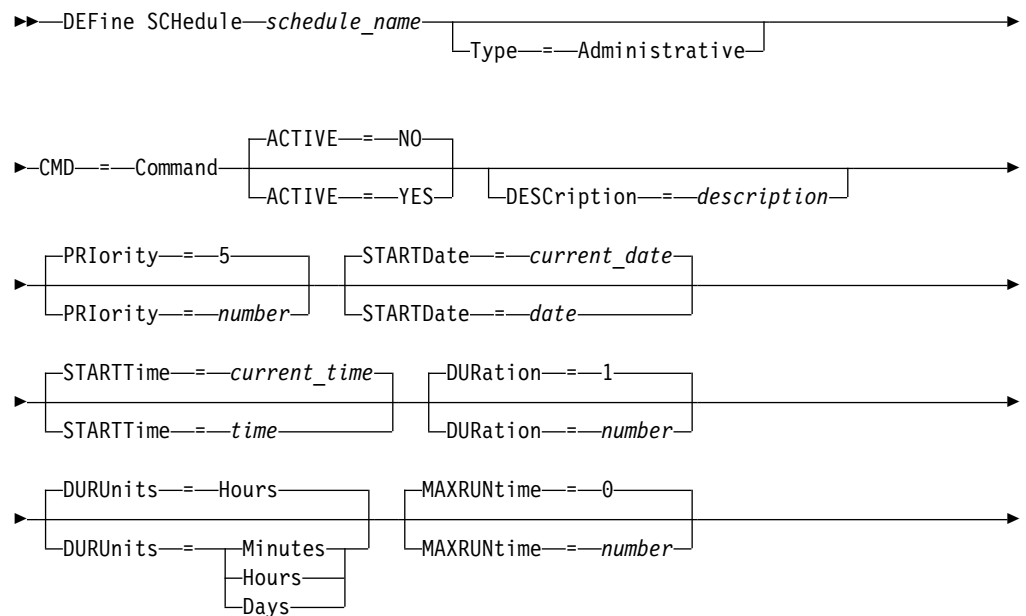
#### Classic administrative schedule

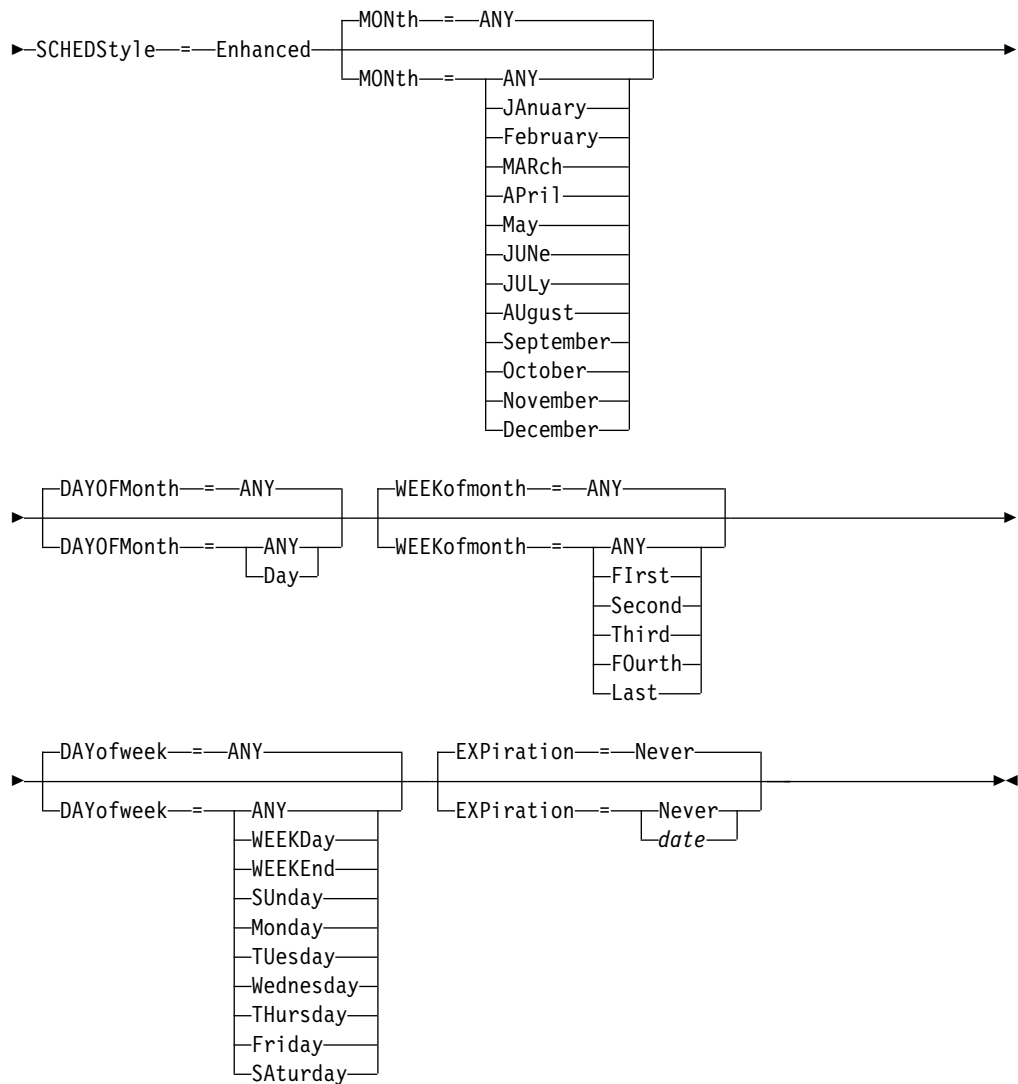




## Syntax

### Enhanced administrative 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.

**Restriction:** 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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
<b>BOTM</b> (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

**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 <b>+HH:MM</b>	The current time plus hours and minutes specified	NOW+02:00 or +02:00.  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 <b>-HH:MM</b>	The current time minus hours and minutes specified	NOW-02:00 or -02:00.  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**.

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. You can specify one of the following values:

#### **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.

### **MAXRUNTime**

Specifies the maximum run time, which is the number of minutes during which server processes that are started by the scheduled commands must be completed. If processes are still running after the maximum run time, the central scheduler cancels the processes.

#### **Tips:**

- The processes might not end immediately when the central scheduler cancels them; they end when they register the cancellation notification from the central scheduler.
- The maximum run time is calculated beginning from when the server process starts. If the schedule command starts more than one process, each process maximum run time is calculated from when the process starts.
- This parameter does not apply to some processes, such as duplicate-identification processes, which can continue to run after the maximum run time.
- This parameter does not apply if the scheduled command does not start a server process.
- Another cancel time might be associated with some commands. For example, the **MIGRATE STGPOOL** command can include a parameter that specifies the length of time that the storage pool migration runs before the migration is automatically canceled. If you schedule a command for which a cancel time is defined, and you also define a maximum run time for the schedule, the processes are canceled at whichever cancel time is reached first.

#### **Restrictions:**

- The value of the parameter is not distributed to servers that are managed by an enterprise configuration manager.
- The value of the parameter is not exported by the **EXPORT** command.

The parameter is optional. You can specify a number in the range 0-1440. The default value is 0. A value of 0 means that the maximum run time is indefinite, and the central scheduler does not cancel processes. The maximum run time must be greater than the startup window duration, which is defined by the **DURATION** and **DURUNITS** parameters.

For example, if the start time of a scheduled command is 9:00 PM, and the duration of the startup window is 2 hours, the startup window is 9:00 PM - 11:00 PM. If the maximum run time is 240 minutes, that is, 4 hours, all applicable server processes that are started by the command must be completed by 1:00 AM. If one or more applicable processes are still running after 1:00 AM, the central scheduler cancels the processes.

**Tip:** Alternatively, you can specify an *end time* of 1:00 AM in the Tivoli Storage Manager Operations Center.

### **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. You can specify one of the following values:

#### **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 was 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 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. You can specify one of the following values:

**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 backup_archivepool type=administrative
cmd="backup stgpool archivepool recoverypool"
schedstyle=enhanced dayofmonth=1,15
```

## DEFINE SCRATCHPADENTRY (Define a scratch pad entry)

Use this command to enter data on a new line in the scratch pad. The scratch pad is a database table that the Tivoli Storage Manager server hosts. You can use the scratch pad to store diverse information in table format.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►—DEFine SCRATCHPadentry—major_category—minor_category—subject—►  
►—Line—=number—Data—=data—►
```

### Parameters

#### *major\_category* (Required)

Specifies the major category in which data is to be stored. Enter a text string of up to 100 alphanumeric characters. This parameter is case sensitive.

#### *minor\_category* (Required)

Specifies the minor category in which data is to be stored. Minor categories are sections within major categories. Enter a text string of up to 100 alphanumeric characters. This parameter is case sensitive.

#### *subject* (Required)

Specifies the subject under which data is to be stored. Subjects are sections within minor categories. Enter a text string of up to 100 alphanumeric characters. This parameter is case sensitive.

#### Line (Required)

Specifies the number of the line on which data is to be stored. Lines are sections within subjects. Specify an integer in the range 1 - 1000.

#### Data (Required)

Specifies the data to be stored on the line. You can enter up to 1000 characters. Enclose the data in quotation marks if the data contains one or more blanks. The data is case sensitive.

### Example: Define a scratch pad entry

Enter the vacation dates of an administrator, Jane, in a table that stores information about the location of all administrators.

```
define scratchpadentry admin_info location jane line=2 data=  
"Out of the office from 1-15 Nov."
```

### Related commands

Table 94. Commands related to DEFINE SCRATCHPADENTRY

Command	Description
DELETE SCRATCHPADENTRY	Deletes a line of data from the scratch pad.
QUERY SCRATCHPADENTRY	Displays information that is contained in the scratch pad.
SET SCRATCHPADRETENTION	Specifies the amount of time for which scratch pad entries are retained.

*Table 94. Commands related to DEFINE SCRATCHPADENTRY (continued)*

<b>Command</b>	<b>Description</b>
UPDATE SCRATCHPADENTRY	Updates data on a line in the scratch pad.

## 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 by using the contents from another script.

The first line for the script can be defined with this command. To add subsequent lines to the script, use the **UPDATE SCRIPT** command.

### Tips:

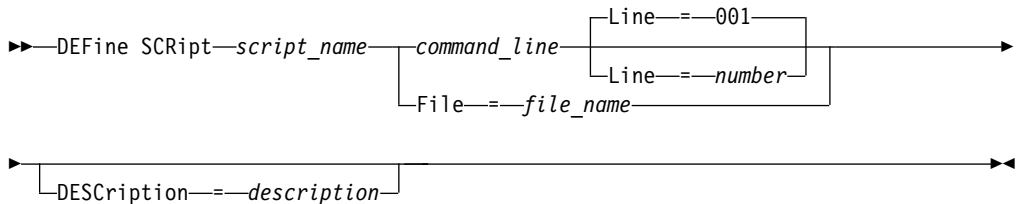
- When routing commands inside scripts, enclose the server or server group in parentheses and omit the colon. Otherwise, if the syntax includes a colon, the command is not routed when the **RUN** command is issued. Instead, the command runs only on the server from which the **RUN** command is issued.
- You cannot redirect the output of a command within a Tivoli Storage Manager script. Instead, run the script and then specify command redirection. For example, to direct the output of **script1** to the `c:\temp\test.out` directory, run the script and specify command redirection as in the following example:

```
run script1 > c:\temp\test.out
```

### 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 that 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.

You can run commands serially, in parallel, or serially and in parallel by specifying the **SERIAL** or **PARALLEL** script commands for the **COMMAND\_LINE** parameter. You can run multiple commands in parallel and wait for them to complete before you proceed to the next command. Commands run serially until the parallel command is encountered.

Conditional logic flow statements can be used. These statements include IF, EXIT, and GOTO.

**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 are read into the script to be defined. The file must reside on the server where this command is running. 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 displays 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 routes 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. The file must be on the server, and be read by the server.

```
define script agadm file=my.script
```

**Related commands**

*Table 95. 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.

*Table 95. Commands related to DEFINE SCRIPT (continued)*

<b>Command</b>	<b>Description</b>
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 to use functions such as virtual volumes, node replication, command routing, and LAN-free data movement, among others.

Use this command to define a server for the following functions:

- Enterprise configuration
- Enterprise event logging
- Command routing
- Virtual volumes
- LAN-free data movement
- Node replication
- Status monitoring of remote servers
- Alert monitoring of remote servers
- Server-to-server export

If you use an LDAP directory server to authenticate passwords, any target servers must be configured for LDAP-authenticated passwords. Data that is replicated from a node that authenticates with an LDAP directory server is inaccessible if the target replication server is not properly configured. If your target replication server is not configured, replicated data from an LDAP node can make it to the target server. But the target replication server must be configured to use LDAP if you want to access the data.

The use of virtual volumes is not supported when the source server and the target server are on the same Tivoli Storage Manager server.

This command also is used to define a Tivoli Storage Manager storage agent as if it were a server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

For:

- Command routing
- Status monitoring of remote servers
- Alert monitoring of remote servers
- Server-to-server export

**Tip:** Command routing uses the ID and the password of the administrator who is issuing the command.

```
►►—DEFine—SERver—server_name—HLAddress—==—ip_address—————►
►—LLAddress—==—tcp_port—————►
   └──COMMmethod—==—TCP/IP┘   └──URL—==—url┘
```

► [DESCRiption==description] ►

## Syntax

For:

- Enterprise configuration
- Enterprise event logging
- Storage agent
- Node replication source and target servers

►►—DEFine—SERver—*server\_name*—SERVERPAssword==—*password*—►

►—HLAddress==—*ip\_address*—LLAddress==—*tcp\_port* [COMMmethod==—TCPiP] ►

► [URL==—*url*] [DESCRiption==—*description*] ►

(1) (2)  
 ► [CROSSDEFine==—No] [VALIdateprotocol==—No] ►  
 [CROSSDEFine==—Yes] [VALIdateprotocol==—All]

► [SSL==—No] ►  
 [SSL==—Yes]

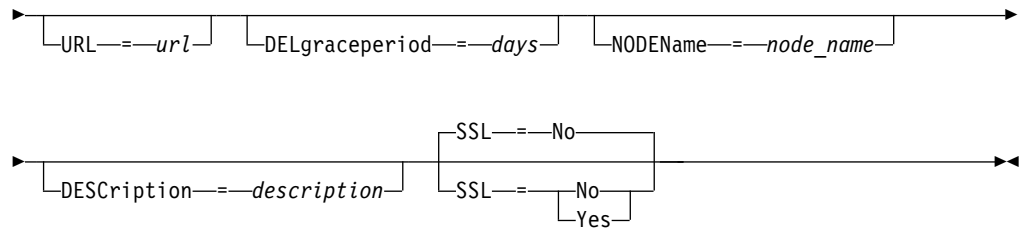
### Notes:

- 1 The CROSSDEFINE parameter does not apply to storage agent definitions.
- 2 The VALIDATEPROTOCOL parameter applies only to storage agent definitions.

## Syntax for virtual volumes

►►—DEFine—SERver—*server\_name*—PAssword==—*password*—►

►—HLAddress==—*ip\_address*—LLAddress==—*tcp\_port* [COMMmethod==—TCPiP] ►



## 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 server-to-server event logging, library sharing, and node replication, you must specify a server name that matches the name that was set by issuing the **SET SERVERNAME** command at the target server.

### **PA**ssword

Specifies the password that is 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.

### **SERVERPA**ssword

Specifies the password of the server you are defining. This password must match the password that is set by the SET SERVERPASSWORD command. This parameter is required for enterprise configuration and server-to-server event logging functions.

### **HL**Address (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.

### **LL**Address (Required)

Specifies the low-level address of the server. This address is usually the same as the address in the TCPPOINT server option of the target server. When SSL=YES, the port must already be designated for SSL communications on the target server.

### **COMM**ethod

Specifies the communication method that is used to connect to the server. This parameter is optional.

### **URL**

Specifies the URL address of this server. The parameter is optional.

### **DEL**graceperiod

Specifies a number of days that an object remains on the target server after it was marked for deletion. You can specify a value 0 - 9999. The default is 5. This parameter is optional.

### **NODEN**ame

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 that is running this command defines itself to the server that is being specified by this command. This parameter is optional.

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.

**Remember:**

- For replication operations, the names of the source and target replication servers must match the names that you specify in this command.
- **CROSSDEFINE** can be used with **SSL=YES** if all of the conditions that are specified for the **SSL=YES** parameter are in place on the source and target server.

You can specify one of the following values:

**No** Cross definition is not completed.

**Yes**

Cross definition is completed.

**VALIDATEPROTOCOL**

Specify a cyclic redundancy check to validate the data that is sent between the storage agent and Tivoli Storage Manager server. The parameter is optional. The default is NO. You can specify one of the following values:

**No** Specifies that data validation is not completed on any data that is sent between the storage agent and server.

**All**

Specifies that data validation is completed 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 affects performance as more resources are required to calculate and compare CRC values between the storage agent and the server.

**SSL**

Specifies the communication mode of the server. The default is NO. You can specify one of the following values:

**No** Specifies that unencrypted communication with the specified server occurs over a TCP/IP session without SSL.

**Yes**

Specifies an SSL session for communication with the specified server. The following conditions apply if you specify YES:

- Before you start the servers, self-signed certificates of the partner servers must be in the key database file (cert.kdb) of each of the servers.
- You can define multiple server names with different parameters for the same target server.
- SSL support is active if the server options file contains the SSLTCPPOINT or SSLTCPADMINPOINT option or if a server is defined with **SSL=YES** at startup.

- Storage agents can issue the **DSMSTA SETSTORAGESEVER** command and include the **STAKEKEYDBPW** and **SSL** parameters to create the key database.
- If third-party certificates are used, the CA (certificate authority) certificate must be kept in the key database of the partner server. If you upgrade from a release earlier than V6.3.0 and have a key database file in your instance directory, update it to allow third-party certificates. To update the key database file, issue the following command:

```
gsk8capicmd_64 -keydb -convert -populate
-db cert.kdb -pw passwordofkeydatabasefile
```

## Example: Set up two servers to use SSL to communicate

You must set up two servers to use SSL to communicate. The server addresses are as follows:

- ServerA is at `bfa.tucson.ibm.com`
- ServerB is at `bfb.tucson.ibm.com`

Complete the following steps to set up the two servers for SSL:

1. Specify options **SSLTCPPORT 1542** and **TCPPORT 1500** for both servers in the `dsmserv.opt` option file.
2. Start both servers and run the following command to set up the key database file password:  

```
SET SSLKEYRINGPW newpw UPDATE=Y
```
3. Shut down both servers to import the `cert256` partner certificate. For ServerA, the certificate is in the `/tsma` instance directory. For ServerB, the certificate is in the `/tsmb` instance directory.
4. Start both servers. The `/tsma/cert256.arm` file is copied to `/tsmb/cert256.bfa.arm` on the `bfb.tucson.ibm.com` address. The `/tsmb/cert256.arm` file is copied to `/tsmb/cert256.bfb.arm` on the `bfa.tucson.ibm.com` address.
5. Issue the following command:

- From ServerA:

```
gsk8capicmd_64 -cert -add -db cert.kdb -pw newpw -format ascii
-label "bfb" -file /tsma/cert256.bfb.arm
```

- From ServerB:

```
gsk8capicmd_64 -cert -add -db cert.kdb -pw newpw -format ascii
-label "bfa" -file /tsmb/cert256.bfa.arm
```

From each server, you can view the certificates in the key database by issuing the following command:

```
gsk8capicmd_64 -cert -list -db cert.kdb -pw newpw
```

6. Restart the servers.
7. Issue the appropriate **DEFINE SERVER** command. For ServerA, issue the following example command:

```
DEFINE SERVER BFB hla=bfb.tucson.ibm.com lla=1542
serverpa=passwordforbfb SSL=YES
```

For ServerB, issue the following example command:

```
DEFINE SERVER BFA hla=bfa.tucson.ibm.com lla=1542
serverpa=passwordforbfa SSL=YES
```

If you do not use SSL, issue the following example **DEFINE SERVER** command on ServerA:

```
DEFINE SERVER BFBTCP hla=bf.b.tucson.ibm.com lla=1500
serverpa=passwordforbfb SSL=NO
```

If you do not use SSL, issue the following example **DEFINE SERVER** command on ServerB:

```
DEFINE SERVER BFATCP hla=bfa.tucson.ibm.com lla=1500
serverpa=passwordforbfa SSL=NO
```

### 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, and set the password to SECRET. Specify that objects remain on the target server for seven days after they are 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 that can receive commands that are routed from other servers. Name the server WEST\_COMPLEX. Set the high-level address to 9.172.12.35, the low-level address to 1500, and the URL address to http://west\_complex:1580/.

```
define server west_complex
hladdress=9.172.12.35 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 96. Commands related to **DEFINE SERVER**

Command	Description
DEFINE DEVCLASS	Defines a device class.
DELETE DEVCLASS	Deletes a device class.

Table 96. Commands related to **DEFINE SERVER** (continued)

<b>Command</b>	<b>Description</b>
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
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 node 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.
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 REPLSERVER	Specifies a target replication server.
UPDATE DEVCLASS	Changes the attributes of a device class.
UPDATE NODE	Changes the attributes that are 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. With a server group, you can route commands to multiple servers by specifying only the group name. After you define 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 97. 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 extra space when predetermined thresholds are exceeded in storage pools that use FILE and DISK device classes. Space triggers are not enabled for storage pools with a parameter RECLAMATIONTYPE=SNAPLOCK.

The Tivoli Storage Manager server allocates more space when space utilization reaches a specified value. After allocating more space, the server 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 can 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.

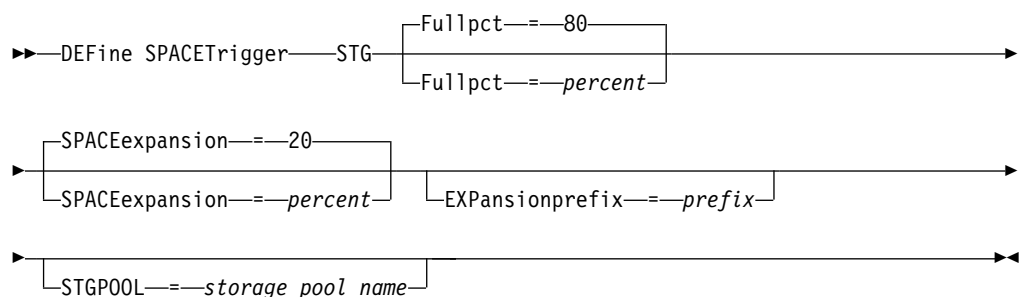
For example, if you specify multiple directories for a device class and the directories reside in the same file system, the server calculates space by adding values representing the space remaining in each directory. These space calculations are inaccurate. Rather than choosing a storage pool with sufficient space for an operation, the server might choose the directory that is specified for the device class 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 parameter is optional. Specify an integer value 0 - 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 might 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 that is 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. This parameter is optional. The default is 20. 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 that are specified with the associated device class.

### **STGPOOL**

Specifies the storage pool that is 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 is 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 98. Commands related to DEFINE SPACETRIGGER*

<b>Command</b>	<b>Description</b>
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 STATUSTHRESHOLD (Define a status monitoring threshold)

Use this command to define a new status monitoring threshold.

Status monitoring thresholds compare the defined conditions to the status monitoring server queries and inserts the results in the status monitoring table.

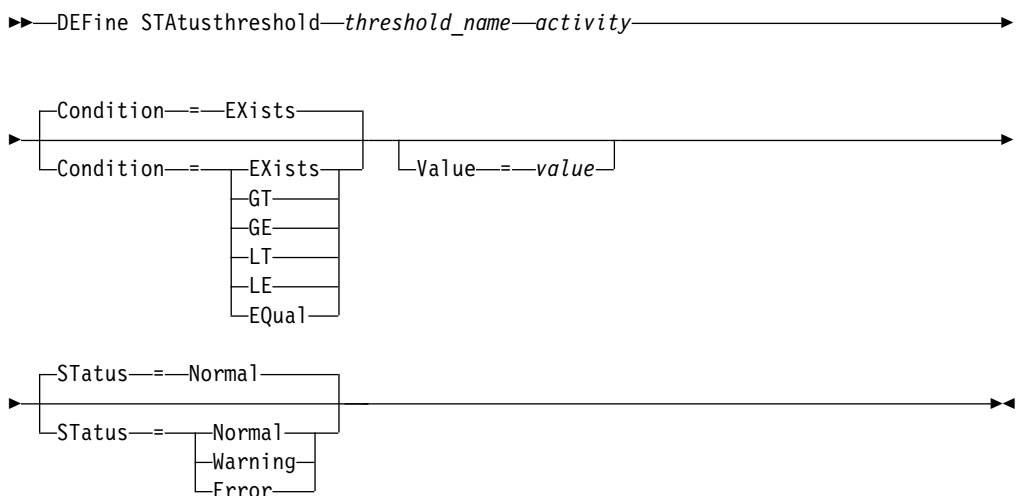
Multiple thresholds can be defined for an activity. For example, you can create a threshold that provides a warning status if storage pool capacity utilization is greater than 80%. You can then create another threshold that provides error status if storage pool capacity utilization is greater than 90%.

**Note:** If a threshold is already defined for an EXISTS condition, you cannot define another threshold with one of the other condition types.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *threshold\_name* (Required)

Specifies the threshold name. The name cannot exceed 48 characters in length.

#### *activity* (Required)

Specifies the activity for which you want to create status indicators. Specify one of the following values:

##### **PROCESSSUMMARY**

Specifies the number of processes that are currently active.

##### **SESSIONSUMMARY**

Specifies the number of sessions that are currently active.

##### **CLIENTSESSIONSUMMARY**

Specifies the number of client sessions that are currently active.

**SCHEDCLIENTSESSIONSUMMARY**

Specifies the number of scheduled client sessions.

**DBUTIL**

Specifies the database utilization percentage. The default warning threshold value is 80%, and the default error threshold value is 90%.

**DBFREESPACE**

Specifies the free space available in the database in gigabytes.

**DBUSEDSPACE**

Specifies the amount of database space that is used, in gigabytes.

**ARCHIVELOGFREESPACE**

Specifies the free space that is available in the archive log, in gigabytes.

**STGPOOLUTIL**

Specifies the storage pool utilization percentage. The default warning threshold value is 80%, and the default error threshold value is 90%.

**STGPOOLCAPACITY**

Specifies the storage pool capacity in gigabytes.

**AVGSTGPOOLUTIL**

Specifies the average storage pool utilization percentage across all storage pools. The default warning threshold value is 80%, and the default error threshold value is 90%.

**TOTSTGPOOLCAPACITY**

Specifies the total storage pool capacity in gigabytes for all available storage pools.

**TOTSTGPOOLS**

Specifies the number of defined storage pools.

**TOTRWSTGPOOLS**

Specifies the number of defined storage pools that are readable or writeable.

**TOTNOTRWSTGPOOLS**

Specifies the number of defined storage pools that are not readable or writeable.

**STGPOOLINUSEANDDEFINED**

Specifies the total number of defined volumes that are in use.

**ACTIVELOGUTIL**

Specifies the current percent utilization of the active log. The default warning threshold value is 80%, and the default error threshold value is 90%.

**ARCHLOGUTIL**

Specifies the current utilization of the archive log. The default warning threshold value is 80%, and the default error threshold value is 90%.

**CPYSTGPOOLUTIL**

Specifies the percent utilization for a copy storage pool. The default warning threshold value is 80%, and the default error threshold value is 90%.

**PMRYSTGPOOLUTIL**

Specifies the percent utilization for a primary storage pool. The default warning threshold value is 80%, and the default error threshold value is 90%.

**DEVCLASSPCTDRVOFFLINE**

Specifies the percent utilization of drives that are offline, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTDRVPOLLING**

Specifies the drives polling, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTLIBPATHSOFFLINE**

Specifies the library paths that are offline, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTPATHSOFFLINE**

Specifies the percentage of device class paths that are offline, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTDISKSNOTRW**

Specifies the percentage of disks that are not writable for the disk device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTDISKSUNAVAILABLE**

Specifies the percentage of the disk volumes that are unavailable, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**FILEDEVCLASSPCTSCRUNALLOCATABLE**

Specifies the percentage of scratch volumes that the server cannot allocate for a given non-shared file device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**Condition**

Specifies the condition that is used to compare the activity output to the specified value. The default value is EXISTS. Specify one of the following values:

**EXists**

Creates a status monitoring indicator if the activity exists.

**GT** Creates a status monitoring indicator if the activity outcome is greater than the specified value.

**GE** Creates a status monitoring indicator if the activity outcome is greater than or equal to the specified value.

**LT** Creates a status monitoring indicator if the activity outcome is less than the specified value.

**LE** Creates a status monitoring indicator if the activity outcome is less than or equal to the specified value.

**EQual**

Creates a status monitoring indicator if the activity outcome is equal to the specified value.

**Value (Required)**

Specifies the value that is compared with the activity output for the specified condition. You must specify this parameter, unless CONDITION is set to EXISTS. You can specify an integer in the range 0 - 999999999999999.

## Status

Specifies that the status indicator created in status monitoring if the condition that is being evaluated passes. This optional parameter has a default value of NORMAL. Specify one of the following values:

### Normal

Specifies that the status indicator has a normal status value.

### Warning

Specifies that the status indicator has a warning status value.

### Error

Specifies that the status indicator has an error status value.

## Define status threshold

Define a status threshold for average storage pool utilization percentage by issuing the following command:

```
define statusthreshold avgstgpl "AVGSTGPOOLUTIL" value=85
  condition=gt status=warning
```

## Related commands

Table 99. Commands related to DEFINE STATUSTHRESHOLD

Command	Description
"DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442	Deletes a status monitoring threshold.
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932	Displays information about a status monitoring thresholds.
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176	Specifies whether to enable client at-risk activity interval evaluation
"SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)" on page 1180	Specifies the refresh interval for status monitoring.
"SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)" on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
"UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389	Changes the attributes of an existing status monitoring threshold.

## DEFINE STGPOOL (Define a storage pool)

Use this command to define a primary storage pool, copy storage pool, an active-data pool, a directory container storage pool, or a container storage pool in a cloud environment.

A primary storage pool provides a destination for backup files, archive files, or files that are migrated from client nodes. A copy storage pool provides a destination for 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. A container storage pool provides a destination for deduplicated files. A cloud storage pool provides storage in a cloud environment. The maximum number of storage pools that you can define for a server is 999.

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, 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.

**Restriction:** If a client is using the simultaneous-write function and data deduplication, the data deduplication feature is disabled during backups to a storage pool.

The DEFINE STGPOOL command takes six forms.

- “DEFINE STGPOOL (Define a primary storage pool assigned to random access devices)” on page 332
- “DEFINE STGPOOL (Define a primary storage pool assigned to sequential access devices)” on page 342
- “DEFINE STGPOOL (Define a copy storage pool assigned to sequential access devices)” on page 360
- “DEFINE STGPOOL (Define an active-data pool assigned to sequential-access devices)” on page 369

The syntax and parameters for each form are defined separately.

*Table 100. Commands related to DEFINE STGPOOL*

Command	Description
BACKUP DB	Backs up the Tivoli Storage Manager database to sequential access volumes.
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.
COPY ACTIVE DATA	Copies active backup data.
DEFINE COLLOC GROUP	Defines a collocation group.
DEFINE COLLOC MEMBER	Adds a client node or file space to a collocation group.
DEFINE DEV CLASS	Defines a device class.
	Defines a storage pool directory to a directory-container or cloud-container storage pool.



Table 100. Commands related to *DEFINE STGPOOL* (continued)

<b>Command</b>	<b>Description</b>
DEFINE VOLUME	Assigns a volume to be used for storage within a specified storage pool.
DELETE COLLOGROUP	Deletes a collocation group.
DELETE COLLOCMEMBER	Deletes a client node or file space 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 COLLOGROUP	Displays information about collocation groups.
QUERY DEVCLASS	Displays information about device classes.
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.
	Repairs a directory-container storage pool.
	Protects a directory-container 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 COLLOGROUP	Updates the description of a collocation group.
UPDATE STGPOOL	Changes the attributes of a storage pool.

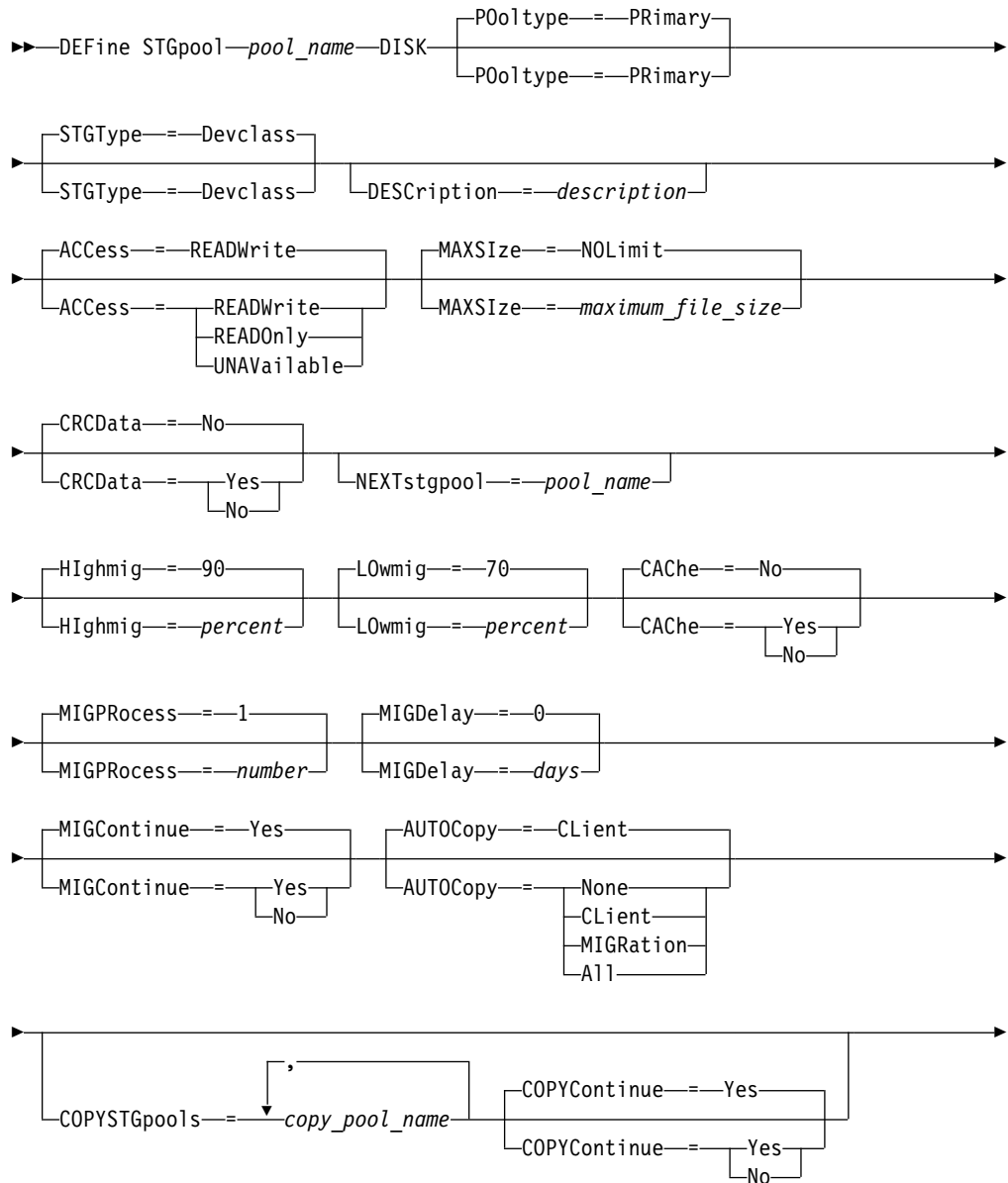
## DEFINE STGPOOL (Define a primary storage pool assigned to random access devices)

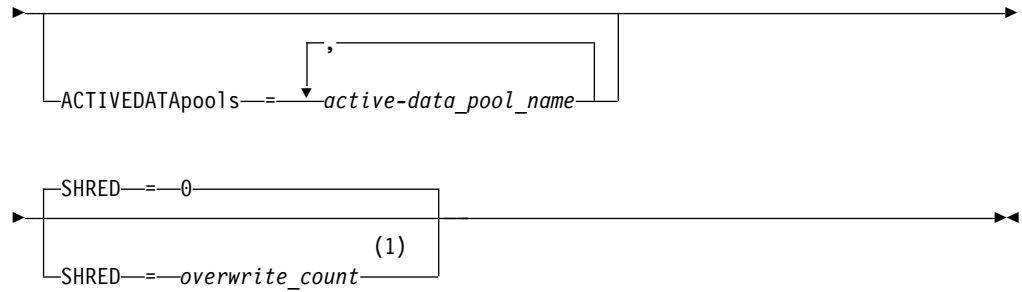
Use this command to define a primary storage pool that is assigned to random access devices.

### 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.

##### **STGType**

Specifies the type of storage that you want to define for a storage pool. This parameter is optional. The default value is DEVCLASS.

##### **Devclass**

Specifies that a device class type of storage pool is assigned to the storage pool.

##### **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. You can specify the following values:

##### **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 was 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 was 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. You can specify the following values:

**NOLimit**

Specifies that there is no maximum size limit for physical files that are stored in the storage pool.

*maximum\_file\_size*

Limits the maximum physical file size. Specify an integer 1 - 999999 terabytes, followed by a scale factor. For example, **MAXSIZE=5G** specifies that the maximum file size for this storage pool is 5 GB. You can use one of the following scale factors:

**Scale factor** **Meaning**

- K kilobyte
- M megabyte
- G gigabyte
- T terabyte

The client estimates the size of files that are sent to the server. The client estimate is used rather than the actual amount of data that is sent to the server. Client options, such as deduplication, compression, and encryption, can cause the actual amount of data that is sent to the server to be larger or smaller than the size estimate. For example, the compression of a file might be smaller in size than the estimate, thus sending less data than the estimate. Furthermore, a binary file might be larger in size after the compression processing, thus sending more data than the estimate.

When the physical size of the storage pool exceeds the **MAXSIZE** parameter, the following table shows where files are typically stored.

*Table 101. The location of a file according to the file size and the pool that is specified.*

File size	Pool specified	Result
Exceeds the maximum size	No pool is specified as the next storage pool in the hierarchy	The server does not store the file
	A pool is specified as the next storage pool in the hierarchy	The server stores the file in the next storage pool that can accept the file size

**Tip:** If you also specify the **NEXTstgpool** parameter, define one storage pool in your hierarchy to have no limit on the maximum file size by specifying the

**MAXSize=NOLimit** parameter. When you have at least one pool with no size limit, you ensure that no matter what its size, the server can store the file.

For multiple files that are sent in a single transaction, the server considers the size of the transaction to be the file size. If the total size of all files in the transaction is larger than the maximum size limit, the server does not store the files in the storage pool.

#### **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 that is stored in your storage hierarchy. You can specify the following values:

##### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more expenditure 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 following actions occur:

- The server cannot migrate files from this storage pool
- The server cannot store files that exceed the maximum size for this storage pool in another storage pool

##### **Restrictions:**

- To ensure that you do not create a chain of storage pools that leads to an endless loop, specify at least one storage pool in the hierarchy with no value.
- If you specify a sequential-access pool as the next storage pool, the pool must be in either NATIVE or NONBLOCK data format.
- Do not specify a directory-container or cloud-container storage pool.
- Do not use this parameter to specify a storage pool for data migration.

#### **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 0 - 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. The **NEXTSTGPOOL** parameter defines this setting. 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 0 - 99. The default value is 70.

When migration is by node or file space, depending upon collocation, the level of the storage pool can fall below the value that you specified for this parameter. To empty the storage pool, set **LOWMIG=0**.

### **CAChe**

Specifies whether the migration process leaves a cached copy of a file in this storage pool after you migrate the file to the next storage pool. This parameter is optional. The default value is **NO**. You can specify the following values:

#### **Yes**

Specifies that caching is enabled.

**No** Specifies that caching is disabled.

Using cache might improve the ability to retrieve files, but might affect the performance of other processes.

### **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 1 - 999. The default value is 1.

During migration, these processes are run in parallel to provide the potential for improved migration rates.

#### **Tips:**

- The number of migration processes is dependent upon the following settings:
  - The **MIGPROCESS** parameter
  - The collocation setting of the next pool
  - The number of nodes or the number of collocation groups with data in the storage pool that is being migrated

For example, suppose that **MIGPROCESS =6**, the next pool **COLLOCATE** parameter is set to **NODE**, but there are only two nodes with data on the storage pool. Migration processing consists of only two processes, not six. If the **COLLOCATE** parameter is set to **GROUP** and both nodes are in the same group, migration processing consists of only one process. If the **COLLOCATE** parameter is set to **NO** or **FILESPACE**, and each node has two file spaces with backup data, then migration processing consists of four processes.

- When you specify this parameter, consider whether the simultaneous-write function is enabled for server data migration. Each migration process requires a mount point and a drive for each copy storage pool and active-data pool that is defined to the target storage pool.

### **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 following items:

- The number of days that the file was in the storage pool
- The number of days, if any, since the file was retrieved by a client

The lesser of the two values are compared to the specified **MIGDELAY** value. For example, if all the following conditions are true, a file is not migrated:

- A file was in a storage pool for five days.
- The file was accessed by a client within the past three days.
- The value that is specified for the **MIGDELAY** parameter is four days.

This parameter is optional. You can specify an integer 0 - 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 that are based 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.

You can specify one of the following values:

##### **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.

#### **AUTOCopy**

Specifies when Tivoli Storage Manager runs simultaneous-write operations. The default value is CLIENT. This parameter is optional and affects the following operations:

- Client store sessions
- Server import processes
- Server data-migration processes

If an error occurs while data is being simultaneously written to a copy storage pool or active-data pool during a migration process, the server stops writing to the failing storage pools for the remainder of the process. However, the server continues to store files into the primary storage pool and any remaining copy storage pools or active-data pools. These pools remain active for the duration of the migration process. Copy storage pools are specified using the **COPYSTGPOOLS** parameter. Active-data pools are specified using the **ACTIVEDATAPOOLS** parameter.

You can specify one of the following values:

##### **None**

Specifies that the simultaneous-write function is disabled.

##### **Client**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions or server import processes.

During server import processes, data is written simultaneously to only copy storage pools. Data is not written to active-data pools during server import processes.

#### **MIGRATION**

Specifies that data is written simultaneously to copy storage pools and active-data pools only during migration to this storage pool. During server data-migration processes, data is written simultaneously to copy storage pools and active-data pools only if the data does not exist in those pools. Nodes whose data is being migrated must be in a domain associated with an active-data pool. If the nodes are not in a domain associated with an active pool, the data cannot be written to the pool.

#### **A11**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions, server import processes, or server data-migration processes. Specifying this value ensures that data is written simultaneously whenever this pool is a target for any of the eligible operations.

#### **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 that are separated by commas. Spaces between the names of the copy pools are not allowed. When you specify a value for the **COPYSTGPools** parameter, you can also specify a value for the **COPYCONTINUE** parameter.

The combined total number of storage pools that are specified in the **COPYSTGPools** 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 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.

The server can write data simultaneously to copy storage pools during the following operations:

- Back up and archive operations by Tivoli Storage Manager backup-archive clients or application clients that are 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

**Restriction:** The simultaneous-write function is not supported for the following store operations:

- When the operation is using LAN-free data movement. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is followed.
- 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 that are defined:
  - The copy storage pools are ignored
  - The data is stored into the primary storage pool only



**Attention:** The function that is provided by the **COPYSTGPOLLS** parameter is not intended to replace the **BACKUP STGPOL** command. If you use the **COPYSTGPOLLS** parameter, continue to use the **BACKUP STGPOL** 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 usually reacts to a copy storage pool write failure for any of the copy storage pools that are listed in the **COPYSTGPOLLS** parameter. This parameter is optional. The default value is YES. When you specify the **COPYCONTINUE** parameter, you must also specify the **COPYSTGPOLLS** parameter.

You can specify the following values:

#### **Yes**

If the **COPYCONTINUE** parameter 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.

**No** If the **COPYCONTINUE** parameter is set to NO, the server will fail the current transaction and discontinue the store operation.

#### **Restrictions:**

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of the active-data pools, the server stops writing to the failing active-data pool for the remainder of the session, but continues storing files into the primary pool and any remaining active-data pools and copy storage pools. The active-data pool list is active only for the life of the session and applies to all the primary storage pools in a particular storage pool hierarchy.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server import. If data is being written simultaneously and a write failure occurs to the primary storage pool or any copy storage pool, the server import process fails.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server data migration. If data is being written simultaneously and a write failure occurs to any copy storage pool or active-data pool, the failing storage pool is removed and the data migration process continues. Write failures to the primary storage pool cause the migration process to fail.

### **ACTIVEDATAPools**

Specifies the names of active-data pools where the server simultaneously writes data during a client backup operation. The **ACTIVEDATAPOLLS** parameter is optional. Spaces between the names of the active-data pools are not allowed.

The combined total number of storage pools that are specified in the **COPYSGTPOLLS** and **ACTIVEDATAPOLLS** 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 that is 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.

The server can write data simultaneously to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients that use 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. Writing data simultaneously to active-data pools is not supported when you use LAN-free data movement. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is followed.
3. The simultaneous-write function 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 that are defined:
  - The active-data pools are ignored
  - The data is stored into the primary storage pool only
4. You cannot use the simultaneous-write function with CENTERA storage devices.
5. Data that is being imported is not 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 that is 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 is physically overwritten when it is deleted. This parameter is optional. You can specify an integer 0 - 10. The default value is 0.

If you specify a value of zero, the server deletes the data from the database. However, the storage that is used to contain the data is not overwritten, and the data exists in storage until that storage is reused for other data. It might be possible to discover and reconstruct the data after it is deleted.

If you specify a value greater than zero, the server deletes the data both logically and physically. The server overwrites the storage that is used to contain the data the specified number of times. This overwriting increases the difficulty of discovering and reconstructing the data after it is deleted.

To ensure that all copies of the data are shredded, specify a **SHRED** value greater than zero for the storage pool that is specified in the **NEXTSTGPOOL** parameter. Do not specify either the **COPYSTGPOOLS** or **ACTIVEDATAPOOLS**. Specifying relatively high values for the overwrite count generally improves the level of security, but might affect performance adversely.

Overwriting of deleted data is done asynchronously after the delete operation is complete. Therefore, the space that is occupied by the deleted data remains occupied for some time. The space is not 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 finishes and identifies files for export, any change 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 canceling 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 that begin 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
```

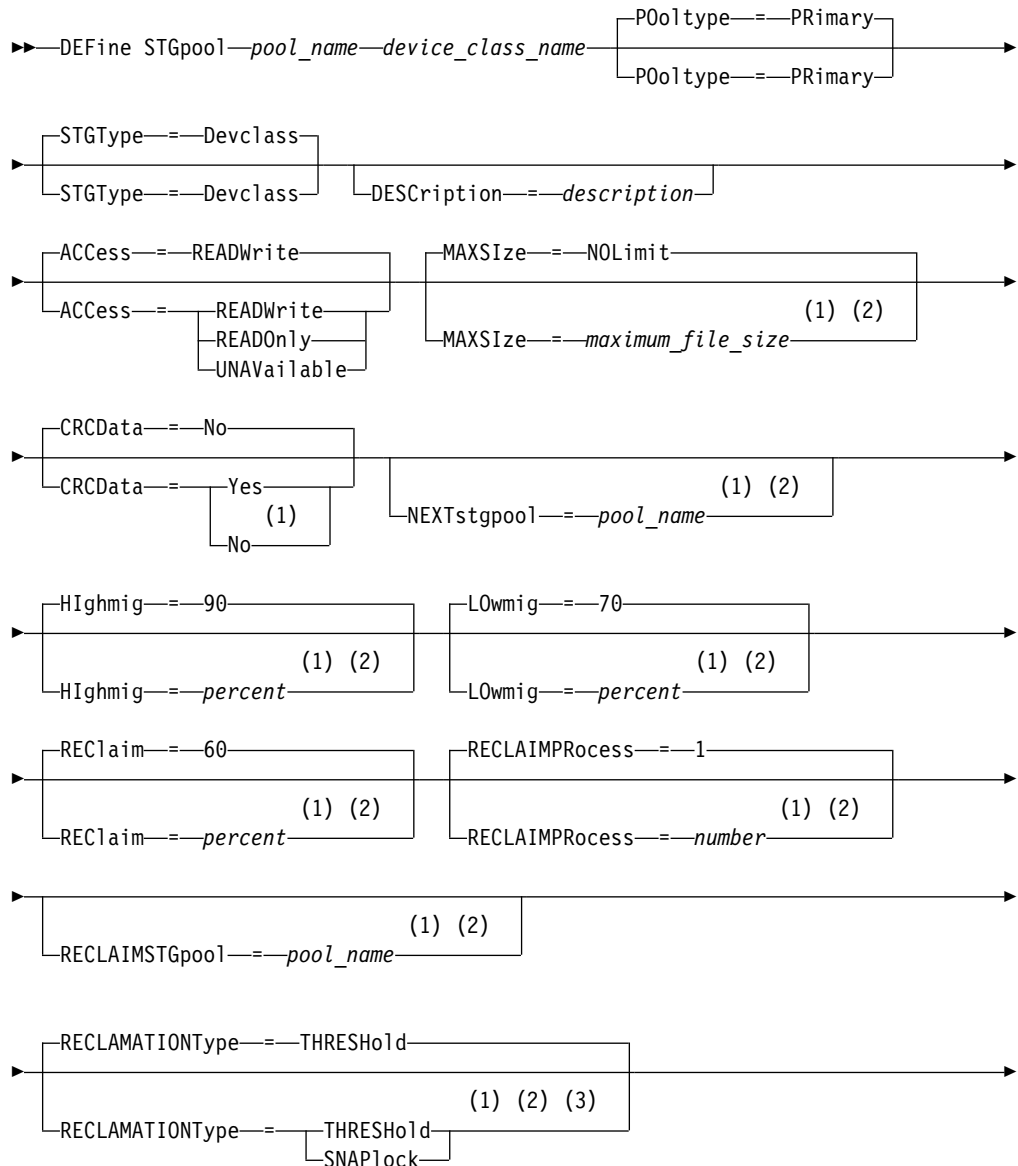
## DEFINE STGPOOL (Define a primary storage pool assigned to sequential access devices)

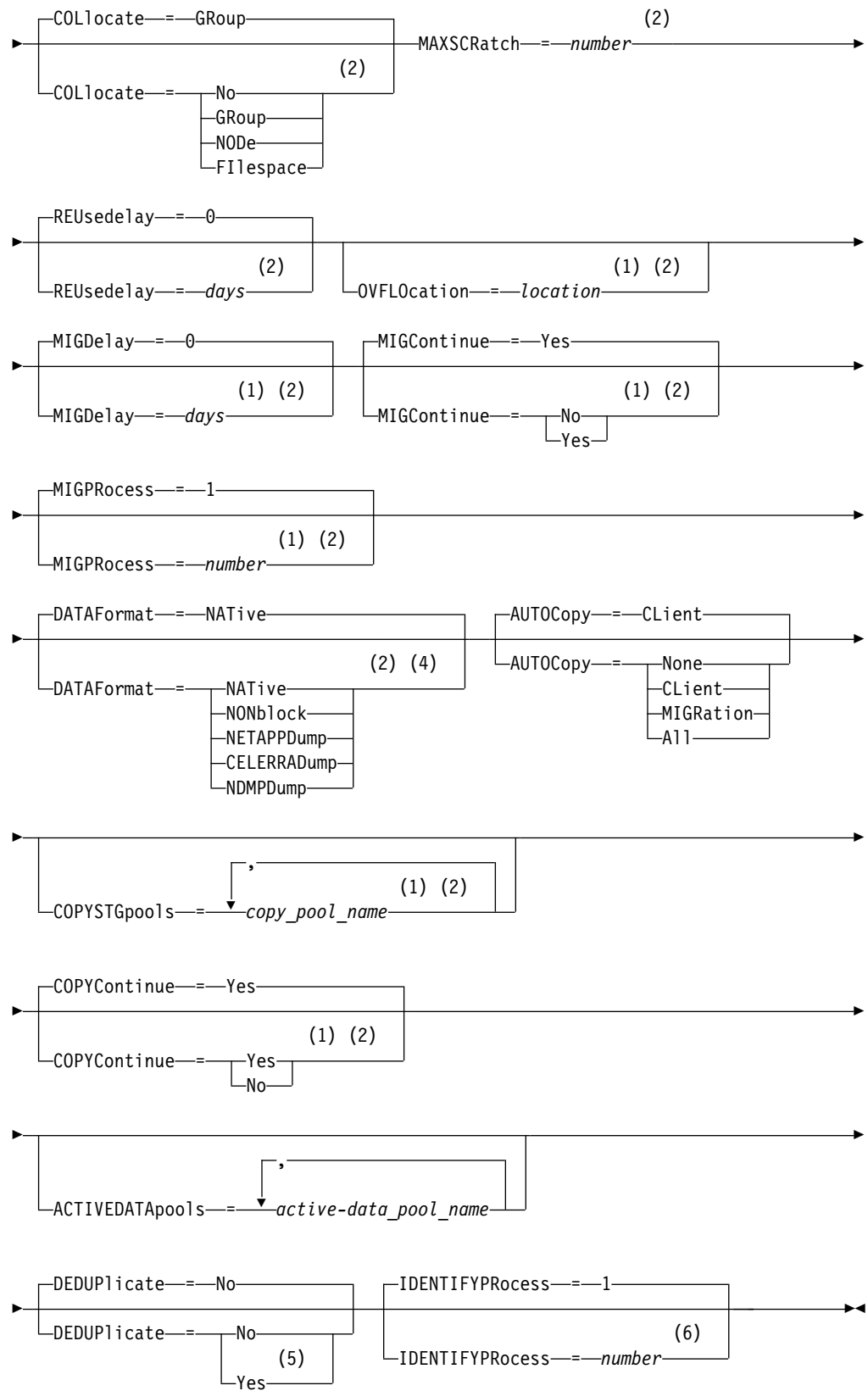
Use this command to define a primary storage pool that is assigned to sequential access devices.

### 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 that are 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 that are specified in the device class must be NetApp SnapLock volumes.
- 4 The values NETAPPDUMP, CELERRADUMP, and NDMPDUMP are not valid for storage pools that are 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.

### **STGType**

Specifies the type of storage that you want to define for a storage pool. This parameter is optional. The default value is DEVCLASS.

### **Devclass**

Specifies that a device class type of storage pool is assigned to the storage pool.

### **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. You can specify the following values:

#### **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 was 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 was 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. You can specify one of the following values:

**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 terabytes, 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

The client estimates the size of files that are sent to the server. The client estimate is used rather than the actual amount of data that is sent to the server. Client options, such as deduplication, compression, and encryption, can cause the actual amount of data that is sent to the server to be larger or smaller than the size estimate. For example, the compression of a file might be smaller in size than the estimate, thus sending less data than the estimate. Furthermore, a binary file might be larger in size after the compression processing, thus sending more data than the estimate.

When the physical size of the storage pool exceeds the **MAXSIZE** parameter, the following table shows where files are typically stored.

*Table 102. The location of a file according to the file size and the pool that is specified*

File size	Pool specified	Result
Exceeds the maximum size	No pool is specified as the next storage pool in the hierarchy	The server does not store the file
	A pool is specified as the next storage pool in the hierarchy	The server stores the file in the next storage pool that can accept the file size

**Tip:** If you also specify the **NEXTstgpool** parameter, define one storage pool in your hierarchy to have no limit on the maximum file size by specifying the **MAXSize=NOLimit** parameter. When you have at least one pool with no size limit, you ensure that no matter what its size, the server can store the file.

For multiple files that are sent in a single transaction, the server considers the size of the transaction to be the file size. If the total size of all files in the transaction is larger than the maximum size limit, the server does not store the files in the storage pool.

**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 that is stored in your storage hierarchy. You can specify the following values:

#### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more processing 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

**Tip:** For storage pools that are associated with the 3592, LTO, or ECARTRIDGE device type, logical block protection provides better protection against data corruption than CRC validation for a storage pool. If you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

To enable logical block protection, specify a value of READWRITE for the **LBPROTECT** parameter on the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands for the 3592, LTO, or ECARTRIDGE device types. Logical block protection is supported only on the following types of drives and media:

- IBM LTO5 and later.
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.
- Oracle StorageTek T10000C and T10000D drives.



### **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 completes 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 might fail because, at the time the actual storage operation is attempted, the space is no longer available.

#### **Restrictions:**

- To ensure that you do not create a chain of storage pools that leads to an endless loop, specify at least one storage pool in the hierarchy with no value.
- If you specify a sequential-access pool as the next storage pool, the pool must be in either **NATIVE** or **NONBLOCK** data format.
- Do not specify a directory-container or cloud-container storage pool.
- Do not use this parameter to specify a storage pool for data migration.
- 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 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 0 - 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 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 0 - 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 allow 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, which is based on the percentage of reclaimable space on a volume. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the database.

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 1 - 100. The default value is 60, except for storage pools that use WORM devices.

The server determines that the volume is a candidate for reclamation if the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool.

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 1 - 999. The default value is 1. You can specify one or more reclamation processes for each primary sequential-access storage pool.

When you calculate the value for this parameter, consider the following resources that are required for reclamation processing:

- The number of sequential storage pools.
- The number of logical and physical drives that can be dedicated to the operation.

To access sequential volumes, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive.

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 that is 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.

**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, the server moves unexpired data from the volumes that are 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. The following are possible values:

##### **THRESHold**

Specifies that volumes that belong to this storage pool are reclaimed based on the threshold value in the RECLAIM attribute for this storage pool.

##### **SNAPlock**

Specifies that FILE volumes that belong to this storage pool are managed for retention using NetApp Data ONTAP software and NetApp SnapLock volumes. This parameter is only valid for storage pools that are 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, which is derived from the retention attributes in the archive copy group for the storage pool, is set in the metadata for the FILE volume by using the SnapLock feature of the NetApp Data ONTAP operating system. Until the retention date expires, 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 that are being defined must be the same when defined to the same device class name. The **DEFINE** command can fail if the **RECLAMATIONTYPE** parameter specified is different from what is defined for storage pools that are 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, which is stored on as few volumes as possible, that belong to one of the following candidates:

- A single client node
- A group of file spaces
- A group of client nodes
- A client file space

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 that is needed to collocate files for storing and the number of volumes required. Collocation can also impact the number of processes migrating disks to sequential pool.

You can specify one of the following options:

**No** Specifies that collocation is disabled. During migration from disk, processes are created at a file space level.

### **GRoup**

Specifies that collocation is enabled at the group level for client nodes or file spaces. For collocation groups, the server attempts to put data for nodes or file spaces that belong to the same collocation group on as few volumes as possible.

If you specify **COLLOCATE=GROUP** but do not define any collocation groups, or if you do not add nodes or file spaces to a collocation group, data is collocated by node. Consider tape usage when you organize client nodes or file spaces into collocation groups.

For example, if a tape-based storage pool consists of data from nodes and you specify **COLLOCATE=GROUP**, the server completes the following actions:

- Collocates the data by group for grouped nodes. Whenever possible, the server collocates data that belongs 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 that are associated with a group.

- Collocates the data by node 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.
- During migration from disk, the server creates migration processes at the collocation group level for grouped nodes, and at the node level for ungrouped nodes.

If a tape-based storage pool consists of data from grouped file spaces and you specify **COLLOCATE=GROUP**, the server completes the following actions:

- Collocates by group, the data for grouped file spaces only. Whenever possible, the server collocates data that belongs to a group of file spaces on a single tape or on as few tapes as possible. Data for a single file space can also be spread across several tapes that are associated with a group.
- Collocates the data by node (for file spaces that are not explicitly defined to a file space collocation group). For example, node1 has file spaces that are named A, B, C, D, and E. File spaces A and B belong to a file space collocation group but C, D, and E do not. File spaces A and B are collocated by file space collocation group, while C, D, and E are collocated by node.
- During migration from disk, the server creates migration processes at the collocation group level for grouped file spaces.

Data is collocated on the least number of sequential access volumes.

#### **NODe**

Specifies that collocation is enabled at the client node level. For collocation groups, 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 try to collocate those file spaces. For compatibility with an earlier version, **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 is collocated by node.

For **COLLOCATE=NODE**, the server creates processes at the node level when you migrate data from disk.

#### **FILESpace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to place data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to place data for different file spaces on different volumes.

For **COLLOCATE=FILESPEC**, the server creates processes at the file space level when you migrate data from disk.

#### **MAXSCRatch (Required)**

Specifies the maximum number of scratch volumes that the server can request for this storage pool. You can specify an integer 0 - 100000000. By allowing the server to request scratch volumes, you avoid having to define each volume to be used.

The value that is 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 use 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 is not reached. The 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 0 - 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.

**Tip:** Use this parameter to 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 that are specified for this parameter must be the same as the number specified for the **SET DRMDBBACKUPEXPIREDAYS** command.

#### **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 0 - 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 that are 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.

You can specify one of the following values:

#### **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 1 - 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 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 161.

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.

**Tip:** When you specify this parameter, consider whether the simultaneous-write function is enabled for server data migration. Each migration process requires a mount point and a drive for each copy storage pool and active-data pool that is defined to the target storage pool.

**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. You can specify the following values:

##### **NATive**

Specifies the data format is the native Tivoli Storage Manager server format and includes block headers.

##### **NONblock**

Specifies the data format is the native Tivoli Storage Manager server format and does not include block headers.

The default minimum block size on a volume that is associated with a FILE device class is 256 KB, regardless how much data is written to the volume. For certain tasks, you can minimize wasted space on storage volumes by specifying the NONBLOCK data format. For example, you can specify the NONBLOCK data format for the following tasks:

- Using content-management products
- Using the DIRMC client option to store directory information
- Migrating very small files by using Tivoli Storage Manager for Space Management or Tivoli Storage Manager HSM for Windows

In most situations, however, the NATIVE format is preferred.

##### **NETAPPDump**

Specifies the data is in a NetApp dump format. This data format must be specified for file system images that are in a dump format and that were backed up from a NetApp or an IBM System Storage N Series file server that uses NDMP. The server does not complete 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 must be reused.

##### **CELERRADump**

Specifies that the data is in an EMC Celerra dump format. This data format must be specified for file system images that are in a dump format and



that were backed up from an EMC Celerra file server that uses NDMP. The server does not complete 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 must be reused.

#### **NDMPDump**

Specifies that the data is in NAS vendor-specific backup format. Use this data format for file system images that were backed up from a NAS file server other than a NetApp or EMC Celerra file server. The server does not complete 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 must be reused.

#### **AUTOCopy**

Specifies when Tivoli Storage Manager completes simultaneous-write operations. The default value is **CLIENT**. This parameter is optional and affects the following operations:

- Client store sessions
- Server import processes
- Server data-migration processes

If the **AUTOCOPY** option is set to **ALL** or **CLIENT**, and there is at least one storage pool that is listed in the **COPYSTGPPOOLS** or **ACTIVEDATAPPOOLS** options, any client-side deduplication is disabled.

If an error occurs while data is being simultaneously written to a copy storage pool or active-data pool during a migration process, the server stops writing to the failing storage pools for the remainder of the process. However, the server continues to store files into the primary storage pool and any remaining copy storage pools or active-data pools. These pools remain active for the duration of the migration process. Copy storage pools are specified using the **COPYSTGPPOOLS** parameter. Active-data pools are specified using the **ACTIVEDATAPPOOLS** parameter.

You can specify one of the following values:

#### **None**

Specifies that the simultaneous-write function is disabled.

#### **Client**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions or server import processes. During server import processes, data is written simultaneously to only copy storage pools. Data is not written to active-data pools during server import processes.

#### **MIGRation**

Specifies that data is written simultaneously to copy storage pools and active-data pools only during migration to this storage pool. During server data-migration processes, data is written simultaneously to copy storage pools and active-data pools only if the data does not exist in those pools. Nodes whose data is being migrated must be in a domain associated with an active-data pool. If the nodes are not in a domain associated with an active pool, the data cannot be written to the pool.

#### **All**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions, server import processes, or

server data-migration processes. Specifying this value ensures that data is written simultaneously whenever this pool is a target for any of the eligible operations.

### **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 that are separated by commas. Spaces between the names of the copy pools are not allowed. When you specify a value for the **COPYSTGPools** parameter, you can also specify a value for the **COPYCONTINUE** parameter.

The combined total number of storage pools that are specified in the **COPYSTGPools** 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 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.

The server can write data simultaneously to copy storage pools during the following operations:

- Back up and archive operations by Tivoli Storage Manager backup-archive clients or application clients that use 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. Writing data simultaneously to copy storage pools is not supported when LAN-free data movement is used. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is accepted.
3. The simultaneous-write function 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 that is provided by the **COPYSTGPools** parameter is not intended to replace the **BACKUP STGPPOOL** command. If you use the **COPYSTGPools** parameter, continue to use the **BACKUP STGPPOOL** 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 reacts to a copy storage pool write failure for any of the copy storage pools that are listed in the **COPYSTGPOLS** parameter. This parameter is optional. The default value is YES. When you specify the **COPYCONTINUE** parameter, you must also specify the **COPYSTGPOLS** parameter.

The **COPYCONTINUE** parameter has no effect on the simultaneous-write function during migration.

You can specify the following values:

#### **Yes**

If the **COPYCONTINUE** parameter 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.

**No** If the **COPYCONTINUE** parameter is set to NO, the server will fail the current transaction and discontinue the store operation.

#### **Restrictions:**

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of the active-data pools, the server stops writing to the failing active-data pool for the remainder of the session, but continues storing files into the primary pool and any remaining active-data pools and copy storage pools. The active-data pool list is active only for the life of the session and applies to all the primary storage pools in a particular storage pool hierarchy.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server import. If data is being written simultaneously and a write failure occurs to the primary storage pool or any copy storage pool, the server import process fails.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server data migration. If data is being written simultaneously and a write failure occurs to any copy storage pool or active-data pool, the failing storage pool is removed and the data migration process continues. Write failures to the primary storage pool cause the migration process to fail.

**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 **ACTIVEDATAPOLS** parameter is optional. Spaces between the names of the active-data pools are not allowed.

The combined total number of storage pools that are specified in the **COPYSGTPOOLS** and **ACTIVEDATAPOLS** 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.

The server can write data simultaneously to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients that use 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. Write data simultaneously to active-data pools is not supported when LAN-free data movement is used. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is accepted.
3. The simultaneous-write function 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 is not 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 that is 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 is 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 server-side duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value 0 - 50. The default value is 1. If the value of the **DEDUPLICATE** parameter is NO, the default setting for IDENTIFYPROCESS has no effect.

**Remember:** 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, becomes idle, and then processes five more files, then the total number of files processed is nine.

Processes end only when canceled or when the number of duplicate-identification processes for the storage pool is changed to a value less than the number currently specified.

**Example: Define a primary storage pool with a TAPE8MM device class**

Define a primary storage pool that is named TAPEPOOL to the TAPE8MM device class (with a device type of GENERICTAPE) with a maximum file size of 5 MB. Store any files larger than 5 MB in subordinate pools, beginning with POOL1. Enable collocation for the storage pool. Allow as many as 5 scratch volumes for this storage pool.

```
define stgpool tapepool tape8mm maxsize=5m
nextstgpool=pool1 collocate=node
maxscratch=5
```



## Notes:

- 1 The RECLAMATIONTYPE=SNAPLOCK setting is valid only for storage pools that are 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 that are 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.

### **STGType**

Specifies the type of storage that you want to define for a storage pool. This parameter is optional. The default value is DEVCLASS.

### **Devclass**

Specifies that a device class type of storage pool is assigned to the 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. You can specify the following values:

#### **READWrite**

Specifies that files can be read from and written to the volumes in the copy storage pool.

#### **READOnly**

Specifies that client nodes can read files that are stored only 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 allowed 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 that are 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 allowed 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, which is stored on as few volumes as possible, that belong to one of the following candidates:

- A single client node
- A group of file spaces
- A group of client nodes
- A client file space

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 that is needed to collocate files for storing and the number of volumes required.

You can specify one of the following options:

**No** Specifies that collocation is disabled.

**GRoup**

Specifies that collocation is enabled at the group level for client nodes or file spaces. For collocation groups, the server attempts to put data for nodes or file spaces that belong to the same collocation group on as few volumes as possible.

If you specify COLLOCATE=GROUP but do not define any collocation groups, or if you do not add nodes or file spaces to a collocation group, data is collocated by node. Consider tape usage when you organize client nodes or file spaces into collocation groups.

For example, if a tape-based storage pool consists of data from nodes and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates the data by group for grouped nodes. Whenever possible, the server collocates data that belongs 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 that are associated with a group.
- Collocates the data by node 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.

If a tape-based storage pool consists of data from grouped file spaces and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates by group, the data for grouped file spaces only. Whenever possible, the server collocates data that belongs to a group of file spaces



on a single tape or on as few tapes as possible. Data for a single file space can also be spread across several tapes that are associated with a group.

- Collocates the data by node (for file spaces that are not explicitly defined to a file space collocation group). For example, node1 has file spaces named A, B, C, D, and E. File spaces A and B belong to a filespace collocation group but C, D, and E do not. File spaces A and B are collocated by filespace collocation group, while C, D, and E are collocated by node.

Data is collocated on the least amount of sequential access volumes.

#### **NODe**

Specifies that collocation is enabled at the client node level. For collocation groups, 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 try to collocate those file spaces. For compatibility with an earlier version, 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 is collocated by node.

#### **Filespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to place data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to place data for different file spaces on different volumes.

#### **RECLaim**

Specifies when the server reclaims a volume, which is based on the percentage of reclaimable space on a volume. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the Tivoli Storage Manager database.

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 1 - 100. The default value is 100, which means that reclamation is not completed.

The server determines that the volume is a candidate for reclamation if the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool.

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 can 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 1 - 999. The default value is 1.

When you calculate the value for this parameter, consider the following resources that are required for reclamation processing:

- The number of sequential storage pools.
- The number of logical and physical drives that can be dedicated to the operation.

To access sequential volumes, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive.

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.

You can specify one or more reclamation processes for each copy storage pool. You can specify multiple concurrent reclamation processes for a single copy storage pool, which makes better use of your available tape drives or FILE volumes. If multiple concurrent processing is not necessary, specify a value of 1 for the **RECLAIMPROCESS** parameter.

### **RECLAMATIONType**

Specifies the method by which volumes are reclaimed and managed. This parameter is optional. The default value is THRESHOLD. The following are possible values:

#### **THRESHold**

Specifies that volumes that belong to this storage pool are reclaimed based on the threshold value in the RECLAIM attribute for this storage pool.

#### **SNAPlock**

Specifies that FILE volumes that belong to this storage pool are managed for retention by using NetApp Data ONTAP software and NetApp SnapLock volumes. This parameter is only valid for storage pools that being defined to a server that has data retention protection that is 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, which is derived from the retention attributes in the archive copy group for the storage pool, is set in the metadata for the FILE volume by using the SnapLock feature of the NetApp Data ONTAP operating system. Until the retention date expires, 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 that are being defined must be the same when defined to the same device class name. The **DEFINE** command fails if the **RECLAMATIONTYPE** parameter specified is different from what is defined for storage pools that are already defined to the device class name.

**OFFSITERECLAIMLimit**

Specifies the number of offsite volumes that space is reclaimed from during reclamation for this storage pool. This parameter is optional. The default value is NOLIMIT. You can specify the following values:

**NOLimit**

Specifies that you want to reclaim the space in all of your offsite volumes.

*number*

Specifies the number of offsite volumes to reclaim space from. You can specify an integer 0 - 99999. A value of zero means that none of the offsite volumes are reclaimed.

**Tip:** To determine the value for the **OFFSITERECLAIMLIMIT**, use the statistical information in the message that is issued at the end of the offsite volume reclamation operation. 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 (Required)**

Specifies the maximum number of scratch volumes that the server can request for this storage pool. You can specify an integer 0 - 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

The value that is 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. An administrator can then query the server for empty, offsite scratch volumes and return them 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 use 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 is not reached. The 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 0 - 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.

**Tip:** Use this parameter to 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 that are specified for this parameter must be the same as the number specified for the **SET DRMDBBACKUPEXPIREDAYS** command.

#### **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. You can specify the following values:

##### **NATive**

Specifies the data format is the native Tivoli Storage Manager server format and includes block headers.

##### **NONblock**

Specifies the data format is the native Tivoli Storage Manager server format and does not include block headers.

The default minimum block size on a volume that is associated with a FILE device class is 256 KB, regardless how much data is written to the volume. For certain tasks, you can minimize wasted space on storage volumes by specifying the NONBLOCK data format. For example, you can specify the NONBLOCK data format for the following tasks:

- Using content-management products
- Using the DIRMC client option to store directory information
- Migrating very small files by using Tivoli Storage Manager for Space Management or Tivoli Storage Manager HSM for Windows

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 were backed up from a NetApp file server by using NDMP. The server does not complete 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 must be reused.

#### **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 were backed up from an EMC Celerra file server by using NDMP. The server does not complete 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 must be reused.

#### **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 were backed up from a NAS file server other than a NetApp or EMC Celerra file server. The server does not complete 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 must be reused.

#### **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 that is stored in your storage hierarchy. You can specify the following values:

##### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more processing is required to calculate and compare CRC values between the storage pool and the server.

**No** Specifies that data is stored without CRC information.

**Tip:** For storage pools that are associated with the 3592, LTO, or **ECARTRIDGE** device type, logical block protection provides better protection against data corruption than CRC validation for a storage pool. If you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

To enable logical block protection, specify a value of **READWRITE** for the **LBPROTECT** parameter on the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands for the 3592, LTO, or **ECARTRIDGE** device types. Logical block protection is supported only on the following types of drives and media:

- IBM LTO5 and later.
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.
- Oracle StorageTek T10000C and T10000D drives.

#### **DEDUPLICATE**

Specifies whether the data that is stored in this storage pool is 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 server-side duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value 0 - 50.

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.

**Remember:** 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, becomes idle, and then processes five more files, then the total number of files processed is nine. Processes end only when canceled or when the number of duplicate-identification processes for the storage pool is changed to a value less than the number 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
```

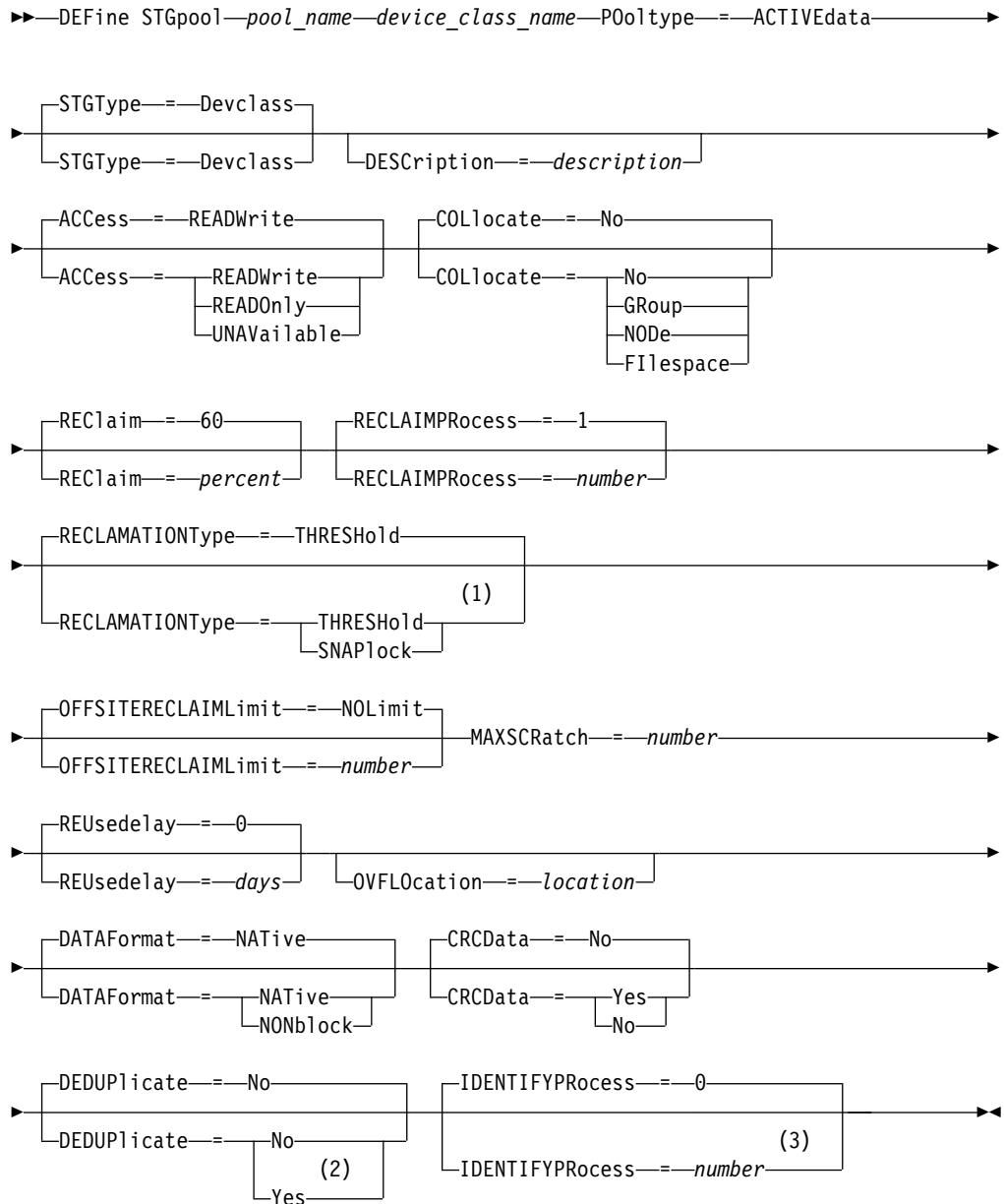
## DEFINE STGPOOL (Define an active-data pool assigned to sequential-access devices)

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



### Notes:

- 1 The `RECLAMATIONTYPE=SNAPLOCK` setting is valid only for storage pools that are 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 that are specified in the device class must be NetApp SnapLock volumes.

- 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.

### **STGType**

Specifies the type of storage that you want to define for a storage pool. This parameter is optional. The default value is DEVCLASS.

### **Devclass**

Specifies that a device class type of storage pool is assigned to the storage 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. You can specify the following values:

#### **READWrite**

Specifies that files can be read from and written to the volumes in the active-data pool.

#### **READOnly**

Specifies that client nodes can read only files that are 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 allowed 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 that are 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 allowed 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, which is stored on as few volumes as possible, that belong to one of the following candidates:

- A single client node
- A group of file spaces
- A group of client nodes
- A client file space

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 that is needed to collocate files for storing and the number of volumes required.

You can specify one of the following options:

**No** Specifies that collocation is disabled.

### **GRoup**

Specifies that collocation is enabled at the group level for client nodes or file spaces. For collocation groups, the server attempts to put data for nodes or file spaces that belong to the same collocation group on as few volumes as possible.

If you specify COLLOCATE=GROUP but do not define any collocation groups, or if you do not add nodes or file spaces to a collocation group, data is collocated by node. Consider tape usage when you organize client nodes or file spaces into collocation groups.

For example, if a tape-based storage pool consists of data from nodes and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates the data by group for grouped nodes. Whenever possible, the server collocates data that belongs 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 that are associated with a group.
- Collocates the data by node 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.

If a tape-based storage pool consists of data from grouped file spaces and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates by group, the data for grouped file spaces only. Whenever possible, the server collocates data that belongs to a group of file spaces on a single tape or on as few tapes as possible. Data for a single file space can also be spread across several tapes that are associated with a group.
- Collocates the data by node (for file spaces that are not explicitly defined to a file space collocation group). For example, node1 has file spaces named A, B, C, D, and E. File spaces A and B belong to a file space

collocation group but C, D, and E do not. File spaces A and B are collocated by file space collocation group, while C, D, and E are collocated by node.

Data is collocated on the least amount of sequential access volumes.

#### **NODe**

Specifies that collocation is enabled at the client node level. For collocation groups, 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 try to collocate those file spaces. For compatibility with an earlier version, 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 is collocated by node.

#### **Filespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to place data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to place data for different file spaces on different volumes.

#### **RECLaim**

Specifies when the server reclaims a volume, which is based on the percentage of reclaimable space on a volume. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the Tivoli Storage Manager database.

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 action makes the original volume available for reuse. This parameter is optional. You can specify an integer 1 - 100. The default value is 60.

The server determines that the volume is a candidate for reclamation if the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool.

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 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 can 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 1 - 999. The default value is 1.

When you calculate the value for this parameter, consider the following resources that are required for reclamation processing:

- The number of sequential storage pools.
- The number of logical and physical drives that can be dedicated to the operation.

To access sequential volumes, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive.

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.

You can specify one or more reclamation processes for each active-data pool. You can specify multiple concurrent reclamation processes for a single active-data pool, which makes better use of your available tape drives or FILE volumes. If multiple concurrent processing is not necessary, specify a value of 1 for the **RECLAIMPROCESS** parameter.

#### **RECLAMATIONType**

Specifies the method by which volumes are reclaimed and managed. This parameter is optional. The default value is THRESHOLD. The following are possible values:

##### **THRESHold**

Specifies that volumes that belong to this storage pool are reclaimed based on the threshold value in the RECLAIM attribute for this storage pool.

##### **SNAPlock**

Specifies that FILE volumes that belong to this storage pool are managed for retention by using NetApp Data ONTAP software and NetApp SnapLock volumes. This parameter is only valid for storage pools that are being defined to a server that has data retention protection that is 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, which is derived from the retention attributes in the archive copy group for the storage pool, is set in the metadata for the FILE volume by using the SnapLock feature of the NetApp Data ONTAP operating system. Until the retention date expires, 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 that are being defined must be the same when defined to the same device class name. The **DEFINE** command fails if the **RECLAMATIONTYPE** parameter specified is different from what is defined for storage pools that are already defined to the device class name.

#### **OFFSITERECLAIMLimit**

Specifies the number of offsite volumes that space is reclaimed from during reclamation for this storage pool. This parameter is optional. The default value is NOLIMIT. You can specify the following values:

##### **NOLimit**

Specifies that you want to reclaim the space in all of your offsite volumes.

*number*

Specifies the number of offsite volumes to reclaim space from. You can specify an integer 0 - 99999. A value of zero means that none of the offsite volumes are reclaimed.

**Tip:** To determine the value for the **OFFSITERECLAIMLIMIT**, use the statistical information in the message that is issued at the end of the offsite volume reclamation operation. 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 are reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 are reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 is reclaimed.

#### **MAXSCRATCH (Required)**

Specifies the maximum number of scratch volumes that the server can request for this storage pool. You can specify an integer 0 - 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

The value that is 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. An administrator can then query the server for empty, offsite scratch volumes and return them 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 use 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 is not reached. The 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 0 - 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.

**Tip:** Use this parameter to 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 that are specified for this parameter must be the same as the number specified for the **SET DRMDBBACKUPEXPIREDAYS** command.

**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. You can specify the following values:

**NATive**

Specifies the data format is the native Tivoli Storage Manager server format and includes block headers.

**NONblock**

Specifies the data format is the native Tivoli Storage Manager server format and does not include block headers.

The default minimum block size on a volume that is associated with a FILE device class is 256 KB, regardless how much data is written to the volume. For certain tasks, you can minimize wasted space on storage volumes by specifying the NONBLOCK data format. For example, you can specify the NONBLOCK data format for the following tasks:

- Using content-management products
- Using the DIRMC client option to store directory information
- Migrating very small files by using Tivoli Storage Manager for Space Management or Tivoli Storage Manager HSM for Windows

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 that is stored in your storage hierarchy. You can specify the following values:

**Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more processing is required to calculate and compare CRC values between the storage pool and the server.

**No** Specifies that data is stored without CRC information.

**Tip:** For storage pools that are associated with the 3592, LTO, or ECARTRIDGE device type, logical block protection provides better protection against data corruption than CRC validation for a storage pool. If you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

To enable logical block protection, specify a value of READWRITE for the **LBPROTECT** parameter on the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands for the 3592, LTO, or ECARTRIDGE device types. Logical block protection is supported only on the following types of drives and media:

- IBM LTO5 and later.
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.
- Oracle StorageTek T10000C and T10000D drives.

#### **DEDUPLICATE**

Specifies whether the data that is stored in this storage pool is 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 server-side duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value 0 - 50.

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.

**Remember:** 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, becomes idle, and then processes five more files, then the total number of files processed is nine. Processes end only when canceled or when the number of duplicate-identification processes for the storage pool is changed to a value less than the number 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 103. 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.

Table 103. Commands related to **DEFINE SUBSCRIPTION** (continued)

<b>Command</b>	<b>Description</b>
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.

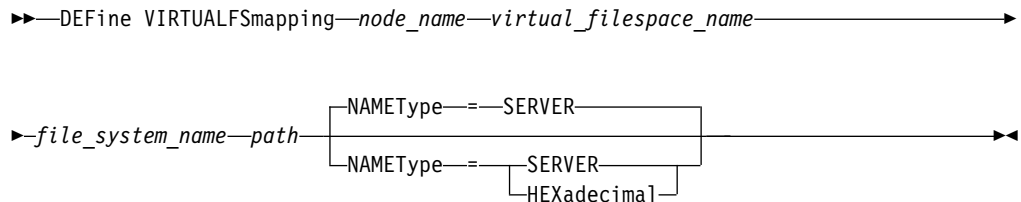
**Note:** The NAS node must have an associated data mover definition because when the Tivoli Storage Manager server updates a virtual file space mapping, the server attempts to contact the NAS device to validate the virtual file system and file system name.

### 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 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 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.

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 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, the server 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 104. 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 you define 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. Alternatively, you can use space triggers to create preassigned volumes when predetermined space-utilization thresholds are exceeded. For details about space triggers, see “DEFINE SPACETRIGGER (Define the space trigger)” on page 323. 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.

**Attention:** Volumes for the z/OS® media server that are created using the **DEFINE VOLUME** command remain physically full or allocated after the server empties the volume, for example, after expiration or reclamation. For FILE volumes, the DASD space is not relinquished to the system when the volume is emptied. If a storage pool requires an empty or filling volume, the FILE volume can be used. In contrast, tape volumes that are logically empty are the same as physically empty. FILE and tape volumes remain defined in the server. In contrast, SCRATCH volumes, including the physical storage that is allocated for SCRATCH FILE volumes, are returned to the system when emptied.

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. Alternatively, you can have the server create volumes that use storage-pool space triggers.

The server 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 after it is defined to the server.

If you change the size of volumes by extending raw logical volumes or otherwise altering the file sizes of the volumes with operating system commands or utilities, the server might not initialize correctly and data can be lost.

### Restrictions:

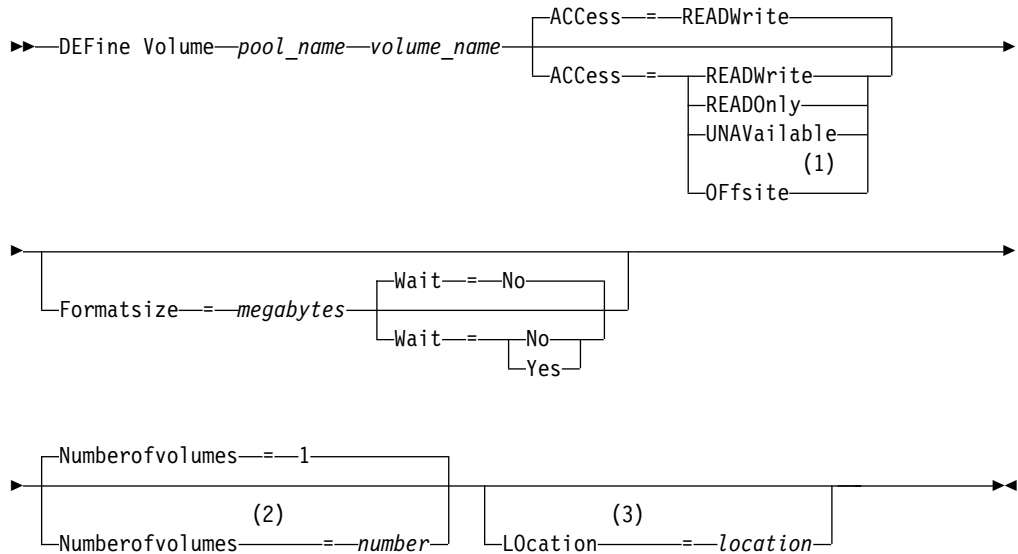
- You cannot use this command to define volumes in storage pools with the parameter setting RECLAMATIONTYPE=SNAPLOCK. Volumes in this type of storage pool are allocated by using the **MAXSCRATCH** parameter on the storage pool definition.
- You cannot define volumes in a storage pool that is defined with the CENTERA device class.

Physical files that are allocated with **DEFINE VOLUME** command are not removed from a file space if you issue the **DELETE VOLUME** 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 assigned.

## Syntax



### Notes:

- 1 This value is valid only for volumes that are 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.

Each volume that is 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.

**Remember:** Volume names cannot contain embedded blanks or equal signs.

See the following tables for volume name requirements:

- Table 105 on page 383: DISK
- Table 106 on page 383: FILE
- Table 107 on page 383: Tape

• Table 108: REMOVABLEFILE

Table 105. Volume name requirements for DISK

Volume Name Requirements	Example
A regular file name, or if you are using a raw logical volume, a character special file. A regular file name can be either a fully qualified path name or a path name relative to the current working directory.	A regular file name with a fully qualified path name: /storage/sbkup01.dsm
<b>Attention:</b> The server cannot always detect if data exists on a raw logical volume. Existing data can be destroyed if you define a volume to that logical volume.	A character mode special file is entered in the format: /dev/vg/rxxx  where <i>vg</i> is the volume group name and <i>xxx</i> is the logical volume name. For example: /dev/groupa/rdsa

Table 106. Volume name requirements for FILE

Volume Name Requirements	Example
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.	/data/fpoo101.dsm
Place FILE volumes in one of the directories that are specified with the DIRECTORY parameter of the <b>DEFINE DEVCLASS</b> command. Otherwise, storage agents might not have access to the volumes. For details, see “DEFINE PATH (Define a path)” on page 262.	

Table 107. Volume name requirements for tape

Volume Name Requirements	Example
Use 1 - 32 alphanumeric characters.	DSMT01
The volume name cannot contain any embedded blanks or equal signs.	

Table 108. Volume name requirements for REMOVABLEFILE

Volume Name Requirements	Example
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 that are stored on the volume.

**UNAVailable**

Specifies that client nodes or server processes cannot access files that are 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 or active-data storage pools.

Use this value to help you track volumes at offsite locations. The server treats volumes that are 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 or active-data 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.

### **Format size**

Specifies the size of the random access volume or FILE volume that is created 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 **FORMATSIZE** parameter is less than or equal to the **MAXCAPACITY** parameter of the **DEFINE DEVCLASS** command.

If you are allocating volumes on a z/OS media server, this parameter is not valid.

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 you specify **WAIT=YES** is specified, the operation is completed 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 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 **FORMATSIZE** parameter. Specify a number from 1 to 256.

If you are allocating volumes on a z/OS media server, the only value that this parameter supports is the default value of 1.

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 and tivoli002. 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.

**Important:** You must ensure that storage agents can access newly created FILE volumes. For more information, see “DEFINE PATH (Define a path)” on page 262.

**Wait**

Specifies whether volume creation and formatting operation is completed in the foreground or background. This parameter is optional. It is ignored unless you also specify the **FORMATSIZE** parameter.

**No** Specifies that a volume creation and formatting operation is completed in the background. The NO value is the default when you also specify a format size.

**Yes**

Specifies that a volume creation and formatting operation is completed in the foreground.

**Remember:** 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 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 that is named FILEPOOL is assigned to a device class with a device type of FILE. Define a volume that is named filepool\_vol01 to this storage pool.

```
define volume filepool /usr/storage/filepool_vol01
```

**Example: Example: Use a background process to define 10 volumes for a file storage pool with a device class 5 GB maximum capacity**

Define 10 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 that is associated with this storage pool is 5 GB. Creation must occur in the background.

```
define volume filepool filevol numberofvolumes=10 formatsize=5000
```

The server creates volume names filevol001 through filevol010.

Volumes are created in the directory or directories that are specified with the `DIRECTORY` parameter of the device class that is 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 109. Commands related to **DEFINE VOLUME***

<b>Command</b>	<b>Description</b>
<code>DEFINE STGPOOL</code>	Defines a storage pool as a named collection of server storage media.
<code>QUERY VOLUME</code>	Displays information about storage pool volumes.
<code>UPDATE DEVCLASS</code>	Changes the attributes of a device class.
<code>UPDATE LIBVOLUME</code>	Changes the status of a storage volume.
<code>UPDATE VOLUME</code>	Updates the attributes of storage pool volumes.



---

## DELETE commands

Use the **DELETE** commands to delete or remove a Tivoli Storage Manager object.

- “DELETE ASSOCIATION (Delete the node association to a schedule)” on page 390
- “DELETE ALERTTRIGGER (Remove a message from an alert trigger)” on page 389
- “DELETE BACKUPSET (Delete a backup set)” on page 392
- “DELETE CLIENTOPT (Delete an option in an option set)” on page 397
- “DELETE CLOPTSET (Delete a client option set)” on page 398
- “DELETE COLLOGROUP (Delete a collocation group)” on page 399
- “DELETE COLLOCMEMBER (Delete collocation group member)” on page 400
- “DELETE COPYGROUP (Delete a backup or archive copy group)” on page 403
- “DELETE DATAMOVER (Delete a data mover)” on page 405
- “DELETE DEVCLASS (Delete a device class)” on page 406
- “DELETE DOMAIN (Delete a policy domain)” on page 407
- “DELETE DRIVE (Delete a drive from a library)” on page 408
- “DELETE EVENT (Delete event records)” on page 409
- “DELETE EVENTSERVER (Delete the definition of the event server)” on page 411
- “DELETE FILESPACE (Delete client node data from the server)” on page 412
- “DELETE GRPMEMBER (Delete a server from a server group)” on page 416
- “DELETE KEYRING (Delete password information in the key database)” on page 417
- “DELETE LIBRARY (Delete a library)” on page 418
- “DELETE MACHINE (Delete machine information)” on page 419
- “DELETE MACHNODEASSOCIATION (Delete association between a machine and a node)” on page 420
- “DELETE MGMTCLASS (Delete a management class)” on page 421
- “DELETE NODEGROUP (Delete a node group)” on page 422
- “DELETE NODEGROUPMEMBER (Delete node group member)” on page 423
- “DELETE PATH (Delete a path)” on page 424
- “DELETE POLICYSET (Delete a policy set)” on page 426
- “DELETE PROFASSOCIATION (Delete a profile association)” on page 427
- “DELETE PROFILE (Delete a profile)” on page 430
- “DELETE RECMEDMACHASSOCIATION (Delete recovery media and machine association)” on page 432
- “DELETE RECOVERYMEDIA (Delete recovery media)” on page 433
- “DELETE SCHEDULE (Delete a client or an administrative command schedule)” on page 434
- “DELETE SCRIPT (Delete command lines from a script or delete the entire script)” on page 438
- “DELETE SERVER (Delete a server definition)” on page 439
- “DELETE SERVERGROUP (Delete a server group)” on page 440
- “DELETE SPACETRIGGER (Delete the storage pool space triggers)” on page 441
- “DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)” on page 442

- “DELETE STGPOOL (Delete a storage pool)” on page 444
- “DELETE SUBSCRIBER (Delete subscriptions from a configuration manager database)” on page 446
- “DELETE SUBSCRIPTION (Delete a profile subscription)” on page 447
- “DELETE VIRTUALFSMAPPING (Delete a virtual file space mapping)” on page 448
- “DELETE VOLHISTORY (Delete sequential volume history information)” on page 449
- “DELETE VOLUME (Delete a storage pool volume)” on page 454

## DELETE ALERTTRIGGER (Remove a message from an alert trigger)

Use this command to remove a message from the list of alert triggers.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
DELEte ALERTTrigger +message_number
```

### Parameters

#### *message\_number* (Required)

Specifies the message number that you want to remove from the list of alert triggers. Specify multiple message numbers, which are separated by commas, and no intervening spaces. Message numbers are a maximum of eight characters in length. Wildcard characters can be used to specify message numbers.

### Delete alert trigger

Delete two message numbers that are designated as alerts, by issuing the following command:

```
delete alerttrigger ANR1067E,ANR1073E
```

### Related commands

Table 110. Commands related to DELETE ALERTTRIGGER

Command	Description
"DEFINE ALERTTRIGGER (Define an alert trigger)" on page 124	Associates specified messages to an alert trigger.
"QUERY ALERTSTATUS (Query the status of an alert)" on page 680	Displays information about alerts that have been issued on the server.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.
"UPDATE ALERTSTATUS (Update the status of an alert)" on page 1205	Updates the status of a reported alert.

## 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*

<b>Command</b>	<b>Description</b>
DEFINE ASSOCIATION	Associates clients with a schedule.
QUERY ASSOCIATION	Displays the clients associated with one or more schedules.

## DELETE BACKUPSET (Delete a backup set)

Use this command to manually delete a backup set before its retention period expires.

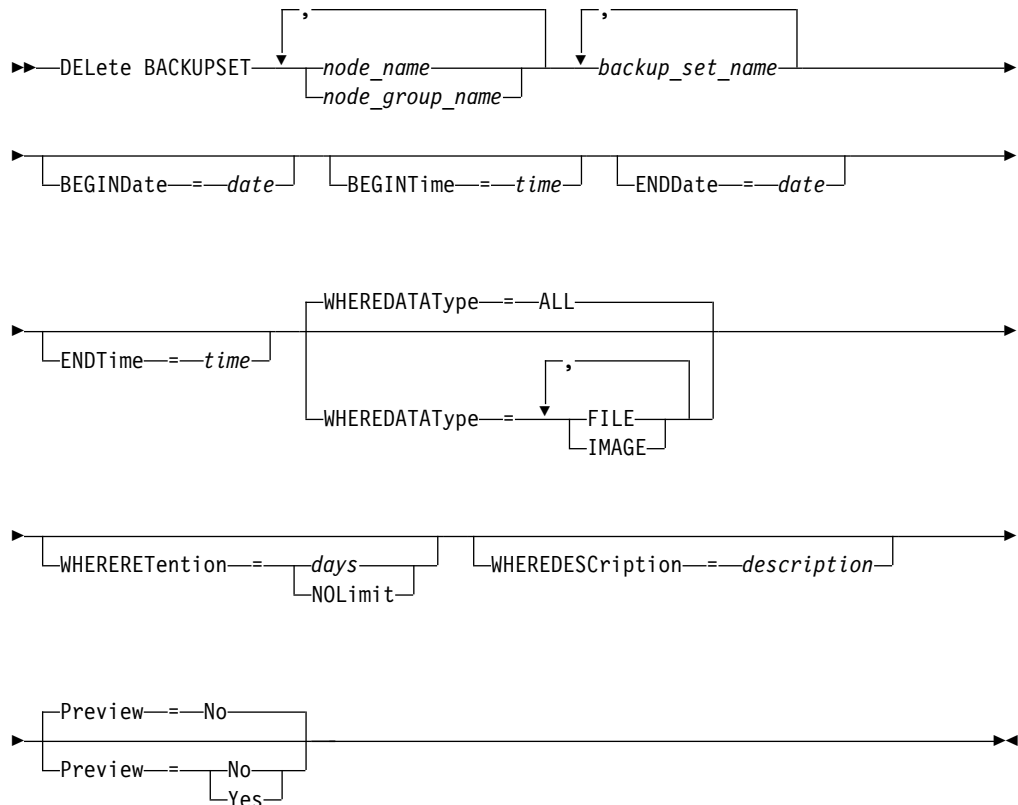
When the server creates a backup set, the retention period assigned to the backup set determines how long the backup set remains in the database. When that date passes, the server automatically deletes the backup set when expiration processing runs. However, you can also manually delete the client's 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
<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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

**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
<i>NOW+HH:MM or +HH:MM</i>	The current time plus hours and minutes specified	NOW+02:00 or +02:00.

Value	Description	Example
NOW- <i>HH:MM</i> or <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.

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 <i>+days</i>	The current date plus days specified.	TODAY +3 or +3.
TODAY- <i>days</i> or <i>-days</i>	The current date minus days specified.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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 <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 <i>-HH:MM</i>	The current time minus hours and minutes on the specified end date	NOW-02:00 or -02:00.



**WHEREDATAType**

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.

**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.
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 CLIENTOPT (Delete an option in an option set)

Use this command to delete a client option in an option set.

### Privilege class

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

### Syntax

```
►►—DELEte CLIENTOpt—option_set_name—option_name—►►
```

└─SEQnumber—=—┬─*number*—┬─

└─ALL—┬─

### Parameters

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

Specifies the name of the client option set.

#### *option\_name* (Required)

Specifies a valid client option.

#### SEQnumber

Specifies a sequence number when an option name is specified more than once. This parameter is optional. Valid values are:

*n* Specifies an integer of 0 or greater.

#### ALL

Specifies all sequence numbers.

### Example: Delete the date format option

Delete the date format option in an option set named *ENG*.

```
delete clientopt eng dateformat
```

### Related commands

Table 113. Commands related to DELETE CLIENTOPT

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 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.

## 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 COLLOGROUP (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 COLLOGGroup—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 colloggroup group1
```

### Related commands

Table 115. Commands related to **DELETE COLLOGGROUP**

Command	Description
DEFINE COLLOGGROUP	Defines a collocation group.
DEFINE COLLOCMEMBER	Adds a client node or file space to a collocation group.
DEFINE STGPOOL	Defines a storage pool as a named collection of server storage media.
DELETE COLLOCMEMBER	Deletes a client node or file space from a collocation group.
MOVE NODEDATA	Moves data for one or more nodes, or a single node with selected file spaces.
QUERY COLLOGGROUP	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 COLLOGGROUP	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 or file space from a collocation group.

### Privilege class

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

### Syntax

#### Delete a node from a collocation group

►► 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 you specify multiple names, separate the names with commas; do not use intervening spaces. You can also use wildcard characters to specify multiple nodes.

#### Delete a file space from a file space collocation group

►► DElete COLLOCMember *group\_name* *node\_name*

►► FIlespace *file\_space\_name* *node\_name*

### Parameters

*group\_name*

Specifies the name of the collocation group from which you want to delete a file space.

*node\_name*

Specifies the client node where the file space is located.

#### FIlespace

Specifies the *file\_space\_name* on the client node that you want to delete from the

collocation group. You can specify one or more file space names that are on a specific client node. If you specify multiple file space names, separate the names with commas, and do not use intervening spaces. You can also use wildcard characters when you specify multiple file space names.

### **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 available only for Windows, Macintosh OS 9, Macintosh OS X, and NetWare. Use this parameter when you specify a file space name that is not a single wildcard. You can specify a fully qualified file space name, which does not have a wildcard. Or you can specify a partly qualified file space name, which can have a wildcard but must contain other characters. The default value is SERVER. Possible values are

#### **SERVER**

The server uses the server 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 code page. Conversion might fail if the string includes characters that are not available in the server code page, or if the server cannot access system conversion routines.

#### **FSID**

The server interprets the file space names by 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 use a single wildcard character for the file space name. The default is BOTH, so the file spaces are included, regardless of code page type. The following values are available:

#### **BOTH**

Include the file spaces, regardless of code page type.

#### **UNICODE**

Include file spaces that are in Unicode only.

#### **NONUNICODE**

Include file spaces that are not in Unicode.

## **Delete collocation group members**

Delete two nodes, NODE1 and NODE2, from a collocation group, GROUP1.

```
delete collocmember group1 node1,node2
```

## **Delete a file space from a file space collocation group**

Issue the following command to delete file space *cap\_27400* from collocation group *collgrp\_2* on node *hp\_4483*:

```
delete collocmember collgrp_2 hp_4483 filespace=cap_27400
```

## Delete a file space collocation group member from a node that uses Unicode

If the file space is on a node that uses Unicode, you can specify that in the command. Issue the following command to delete file space *cap\_257* from collocation group *collgrp\_3* from the *win\_4687* node:

```
delete collocmember collgrp_3 win_4687 filespace=cap_257 codetype=unicode
```

## Delete a file space with a partial name designated

If the file space has a partial name, you can use a wildcard to delete it. Issue the following command to delete file space *cap\_* from collocation group *collgrp\_4* from *win\_4687* node:

```
delete collocmember collgrp_4 win_4687 filespace=cap_* codetype=unicode
```

If there is more than one file space whose name begins with *cap\_*, those file spaces are also deleted.

## Related commands

Table 116. Commands related to **DELETE COLLOCMEMBER**

Command	Description
DEFINE COLLOGROUP	Defines a collocation group.
DEFINE COLLOCMEMBER	Adds a client node or file space 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 FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the 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.
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.

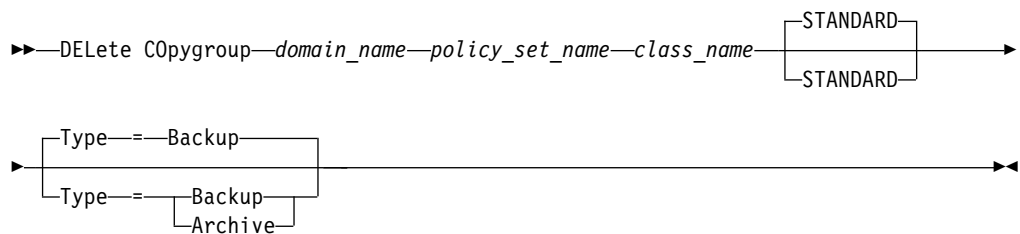
When you activate the changed policy set, any files that are bound to a deleted copy group are managed by the default management class.

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 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 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, including the ACTIVE policy set, management classes, and copy groups are deleted along with the policy domain.

You cannot delete a policy domain to which client nodes are registered. To determine if any client nodes are registered to a policy domain, issue the **QUERY DOMAIN** or the **QUERY NODE** command. Move any client nodes to another policy domain, or delete the nodes.

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.
PERFORM LIBACTION	Defines all drives and paths for a library.
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.

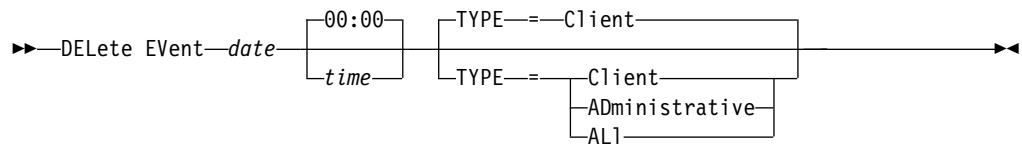
This command only deletes the event records that exist at the time the command is processed. An event record will not be found:

- 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
<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-3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### *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
NOW+HH:MM or +HH:MM	The current time plus hours and minutes specified	NOW+03:00 or +03:00
NOW-HH:MM or -HH:MM	The current time minus hours and minutes specified	NOW-03:00 or -03:00

**Attention:** 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.

### **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, and any file space collocation groups.

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 is returned to scratch if it has space on which data can be written. (Data on WORM volumes, including deleted and expired data, cannot be overwritten. Therefore, data can be written only in space that does not contain current, deleted, or expired data.) If a WORM volume does not have any space available on which data can be written, it remains private. To remove the volume from the library, you must use the **CHECKOUT LIBVOLUME** command.

### Tips:

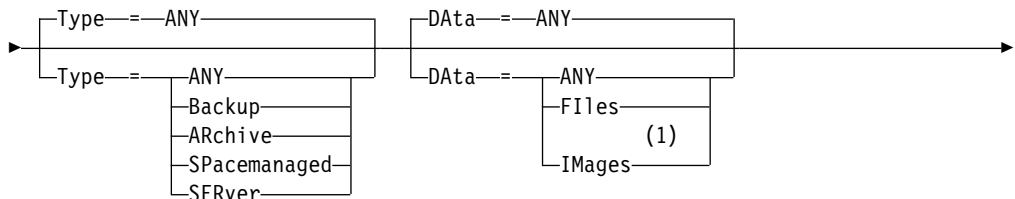
- If archive retention protection is enabled, the server deletes archive files with expired retention periods. For more information, see the **SET ARCHIVERETENTIONPROTECTION** command.
- The server does not delete archive files that are on deletion hold until the hold is released.
- Reclamation does not start while the **DELETE FILESPACE** command is running.
- If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
- If you delete a file space in a deduplicated storage pool, the file space name DELETED is displayed in the output of the **QUERY OCCUPANCY** command until all deduplication dependencies are removed.

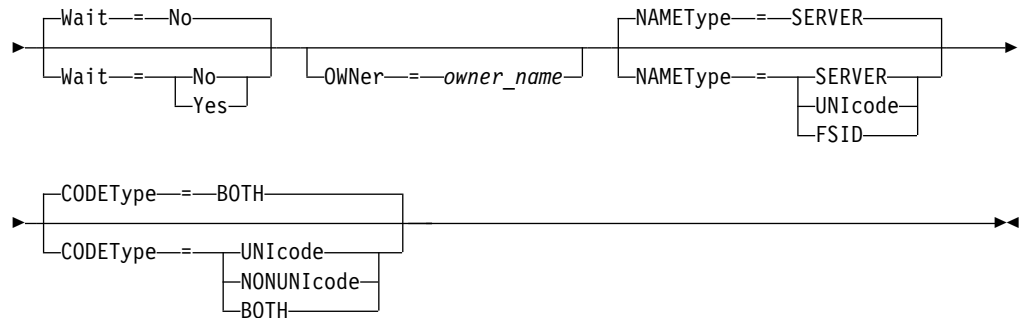
### Privilege class

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—*node\_name*—*file\_space\_name*—►►





**Notes:**

- 1 This parameter can be used only 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 might have the server convert the file space name that you enter. For example, you might want to have the server convert the name that you entered 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. You can use the following values:

**ANY**

Delete only backed-up versions of files and archived copies of files.

If you specify delete file\_space *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 moved 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 that are 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. You can specify one of the following values:

**ANY**

Delete files, directories, and images.

**Files**

Delete files and directories.

**Images**

Delete image objects. You can use this parameter only if you 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. You can specify one of the following values:

**No** Specifies that the server processes this command in the background. You can continue with other tasks while the command is being processed.

Messages that are 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 you continue 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 that belong to the owner. This parameter is optional; it is ignored when TYPE=SPACEMANAGED. This parameter applies to only multiuser client systems such as AIX, Linux, and Solaris 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 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. You can specify one of the following values:

**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 cannot access 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. You can specify one of the following values:

**UNICODE**

Include file spaces that are in Unicode.

**NONUNICODE**

Include file spaces that are not in Unicode.

**BOTH**

Include file spaces regardless of code page type.

**Delete a file space**

Delete the C\_Drive file space that belongs to the client node HTANG.

```
delete filespace htang C_Drive
```

**Delete all space-managed files for a client node**

Delete all files that are migrated from client node APOLLO (that is, all space-managed files).

```
delete filespace 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 filespace 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 key database)

Use this command to delete the password information in the Tivoli Storage Manager database for the key database (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 the cert.kdb file 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).

If the password information is lost after it was updated outside of the server, use this command to delete the key database file information from the server database. You can also delete cert.\* files from the server instance directory. When the server is restarted, it regenerates the cert.kdb file.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—DELEte KEYRING—◄◄

### Parameters

None

### Example: Delete password information in the 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

Table 126. Commands related to DELETE KEYRING

Command	Description
QUERY SSLKEYRINGPW	Displays the Secure Sockets Layer (SSL) key database file password.
SET SSLKEYRINGPW	Sets or updates the key database file password.

## 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 127. 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.
PERFORM LIBACTION	Defines all drives and paths for 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.



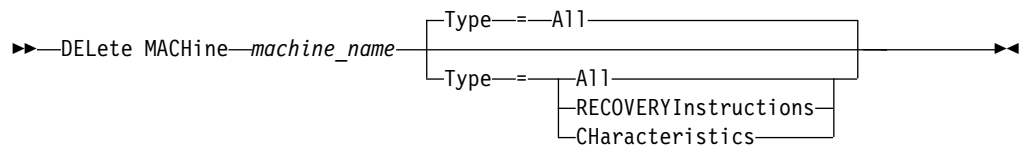
## DELETE MACHINE (Delete machine information)

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 128. Commands related to **DELETE MACHINE**

Command	Description
DEFINE MACHINE	Defines a machine for DRM.
INSERT MACHINE	Inserts machine characteristics or recovery instructions into the 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 129. Commands related to DELETE MACHNODEASSOCIATION

Command	Description
DEFINE MACHNODEASSOCIATION	Associates a 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 130. 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 131. 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 132. 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.



## Related commands

Table 133. Commands related to **DELETE PATH**

Command	Description
DEFINE DATAMOVER	Defines a data mover to the Tivoli Storage Manager server.
DEFINE PATH	Defines a path from a source to a destination.
PERFORM LIBACTION	Defines all drives and paths for a library.
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. When you delete a policy set, all management classes and copy groups that belong to the policy set are also deleted.

The ACTIVE policy set in a policy domain cannot be deleted. You can replace the contents of the ACTIVE policy set by activating a different policy set. Otherwise, the only way to remove the ACTIVE policy set is to delete the policy domain that contains the policy set.

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 by issuing the following command:

```
delete policyset employee_records vacation
```

### Related commands

Table 134. 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)

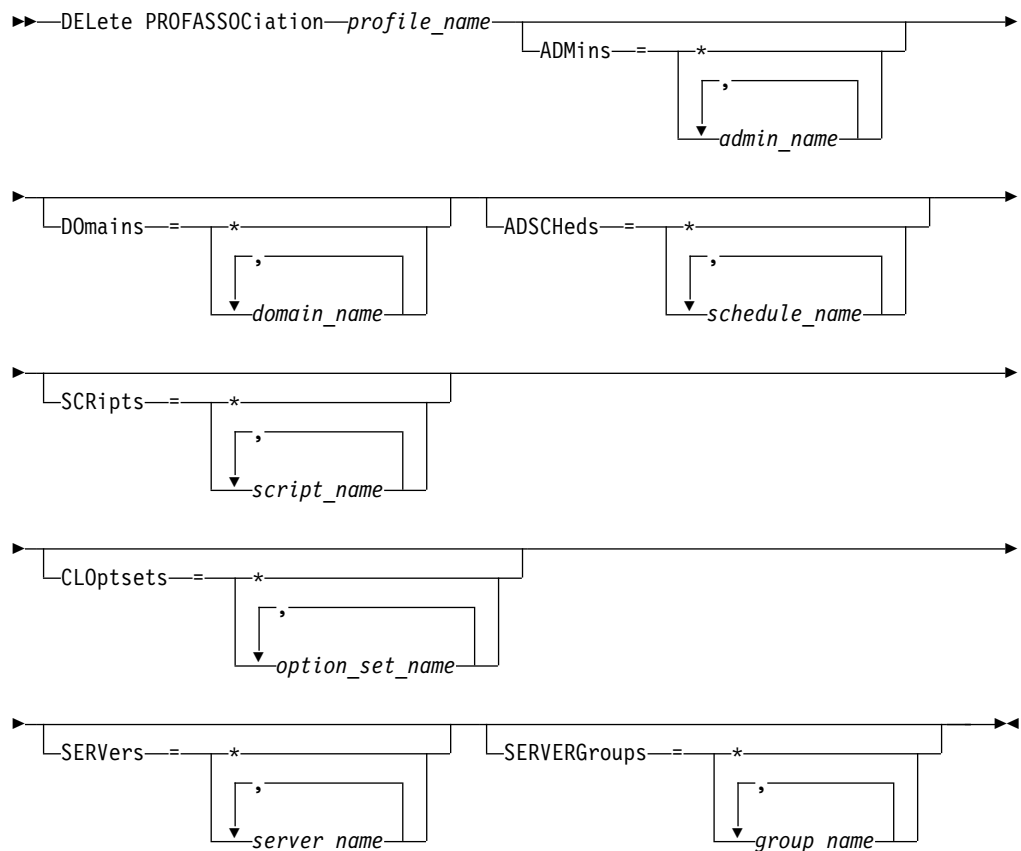
Use this command on a configuration manager to delete the association of one or more objects from a profile. If associations are deleted, the objects are no longer distributed to subscribing managed servers. When managed servers request updated configuration information, the configuration manager notifies 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:

- 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 135. 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.

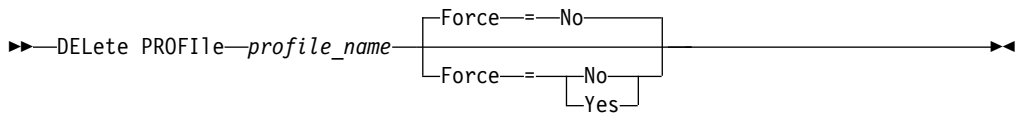
You cannot delete a locked profile. You must first unlock the profile with the **UNLOCK PROFILE** command.

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 136. Commands related to **DELETE 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.

Table 136. Commands related to **DELETE PROFILE** (continued)

<b>Command</b>	<b>Description</b>
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 137. 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 138. 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.

- “DELETE SCHEDULE (Delete an administrative schedule)” on page 436
- “DELETE SCHEDULE (Delete a client schedule)” on page 435

*Table 139. 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 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 SCRATCHPADENTRY (Delete a scratch pad entry)

Use this command to delete one or more lines of data from a scratch pad.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

► `DELEte SCRATCHPadentry—major_category—minor_category—subject` ►



### Parameters

#### *major\_category* (Required)

Specifies the major category from which one or more lines of data are to be deleted. This parameter is case sensitive.

#### *minor\_category* (Required)

Specifies the minor category from which one or more lines of data are to be deleted. This parameter is case sensitive.

#### *subject* (Required)

Specifies the subject from which one or more lines of data are to be deleted. This parameter is case sensitive.

#### **Line**

Specifies a line of data that is to be deleted. For *number*, enter the number of the line that is to be deleted. All data on the line is deleted. The numbering of other lines in the subject section is not affected. You can delete all lines of data from a subject section by omitting the **Line** parameter in this command.

### Example: Delete all lines of data from a subject in a scratch pad

Delete all lines of data about the location of an administrator, Jane, from a database that stores information about administrators:

```
delete scratchpadentry admin_info location jane
```

### Related commands

Table 140. Commands related to DELETE SCRATCHPADENTRY

Command	Description
DEFINE SCRATCHPADENTRY	Creates a line of data in the scratch pad.
QUERY SCRATCHPADENTRY	Displays information that is contained in the scratch pad.
SET SCRATCHPADRETENTION	Specifies the amount of time for which scratch pad entries are retained.
UPDATE SCRATCHPADENTRY	Updates data on a line in the scratch pad.

## 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 qsample line=5
```

### Related commands

Table 141. Commands related to DELETE SCRIPT

Command	Description
COPY SCRIPT	Creates a copy of a script.
DEFINE SCRIPT	Defines a script to the 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 142. 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 143. 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 144. 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 STATUSTHRESHOLD (Delete a status monitoring threshold)

Use this command to delete an existing status monitoring threshold.

Status monitoring thresholds compare the defined conditions to the status monitoring server queries and inserts the results in the status monitoring table.

Multiple thresholds can be defined for an activity. For example, you can create a threshold that provides a warning status if storage pool capacity utilization is greater than 80%. You can then create another threshold that provides error status if storage pool capacity utilization is greater than 90%.

**Note:** If a threshold is already defined for an EXISTS condition, you cannot define another threshold with one of the other condition types.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—DELEte STATusthreshold—threshold_name—————►►
```

### Parameters

*threshold\_name* (**Required**)

Specifies the threshold name that you want to delete.

### Delete an existing status threshold

Delete an existing status threshold by issuing the following command:

```
delete statusthreshold avgstgpl
```

### Related commands

Table 145. Commands related to DELETE STATUSTHRESHOLD

Command	Description
"DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)" on page 326	Defines a status monitoring threshold.
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932	Displays information about a status monitoring thresholds.
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)" on page 1180	Specifies the refresh interval for status monitoring.



Table 145. Commands related to **DELETE STATUSTHRESHOLD** (continued)

Command	Description
"UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389	Changes the attributes of an existing status monitoring threshold.

## 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 that are 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 about storage pool hierarchy, see the **NEXTSTGPOOL** parameter in the **DEFINE STGPOOL** command.

### Restrictions:

- 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 you delete a copy storage pool that was previously included in a primary storage-pool definition (specifically in the COPYSTGPOOLS list), you must remove the copy storage pool from the list before deletion. Otherwise, the **DELETE STGPOOL** command fails until all references to that copy pool are 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 146. 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.
	Defines a storage pool directory to a directory-container or cloud-container storage pool.
	Deletes a storage pool directory from a directory-container or cloud-container storage pool.

Table 146. Commands related to **DELETE STGPOOL** (continued)

<b>Command</b>	<b>Description</b>
QUERY STGPOOL	Displays information about storage pools.
	Displays information about storage pool directories.
SET DRMCOPYSTGPOOL	Specifies that copy storage pools are managed by DRM.
UPDATE STGPOOL	Changes the attributes of a storage pool.
	Changes the attributes of a storage pool directory.

## 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 147. 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.

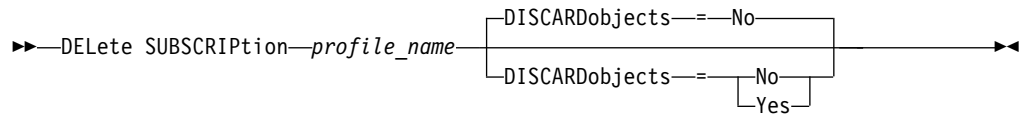
## DELETE SUBSCRIPTION (Delete a profile subscription)

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 148. 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 149. 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:

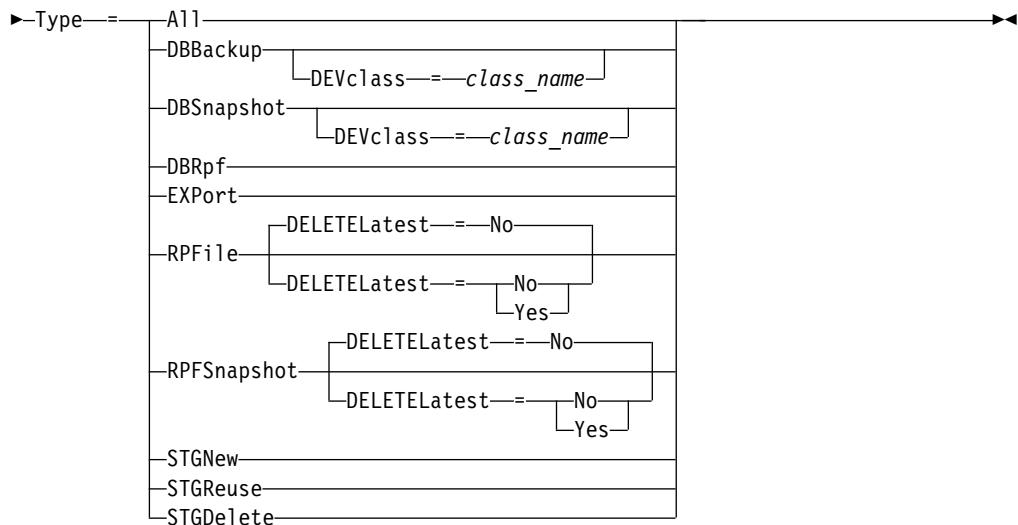
1. Volumes for the most recent database backup series are not deleted.
2. Existing volume history files are not automatically updated with this command.
3. You can use the DEFINE SCHEDULE command to periodically delete volume history records.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
▶▶ DElete VOLHistory TDate = 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
<i>MM/DD/YYYY</i>	A specific date	01/23/1999
TODAY	The current date	TODAY
<i>TODAY-days</i> or <i>-days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY-30 or -30.  To delete records that are 30 or more days old, you can specify TODAY-30 or simply -30.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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
<i>HH:MM:SS</i>	A specific time on the specified date	12:30:22
NOW	The current time on the specified date	NOW
NOW+ <i>HH:MM</i> or <i>+HH:MM</i>	The current time plus hours and minutes on the specified date	NOW+03:00 or +03:00.  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- <i>HH:MM</i> or <i>-HH:MM</i>	The current time minus hours and minutes on the specified date	NOW-03:30 or -03:30.  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:

##### A11

Specifies to delete all records.

**Note:** The **DELETE VOLHISTORY** command does not delete records of remote 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.

**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.

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 150. 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 onsite and offsite.
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 you issue the **DELETE VOLUME** command, volume information is deleted from the Tivoli Storage Manager database. However, the physical files that are allocated with **DEFINE VOLUME** command are not removed from the file space.

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.

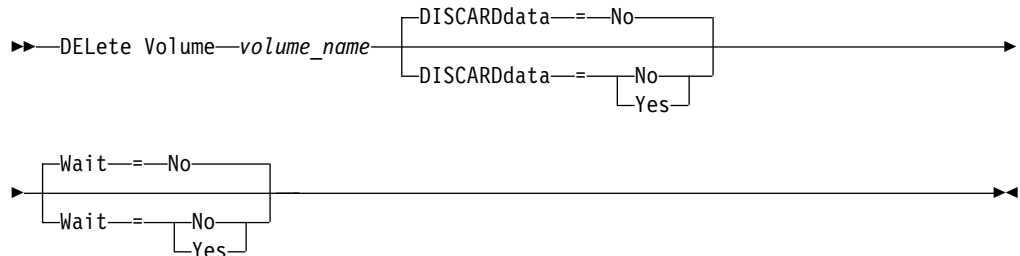
If you issue the **DELETE VOLUME** command for a volume in a storage pool that has a **SHRED** parameter value greater than 0, the volume is placed in the pending state until shredding is run. Shredding is necessary to complete the deletion, even if the volume is empty.

If you issue the **DELETE VOLUME** command for a volume in a storage pool that is set up for data deduplication, the server destroys any object that is referencing data on that 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.

### **Remember:**

1. The server does not delete archive files that are on deletion hold.
2. If archive retention protection is enabled, the 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.

**Remember:** 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 151. 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 onsite and offsite.
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.

- “DISABLE EVENTS (Disable events for event logging)” on page 458
- “DISABLE REPLICATION (Prevent outbound replication processing on a server)” on page 462
- “DISABLE SESSIONS (Prevent new sessions from accessing Tivoli Storage Manager)” on page 463

## 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:

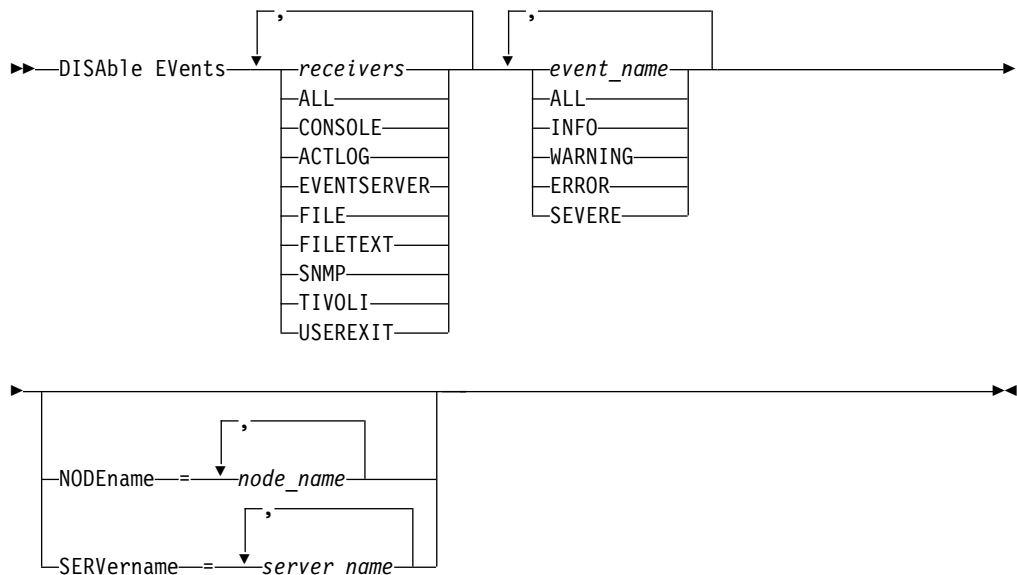
- Certain messages are displayed on the console even if they are disabled. These include some messages issued during server startup and shutdown and responses to administrative commands.
- Server messages from the server on which this command is issued cannot be disabled for the activity log.

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®	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 152. 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.

Table 152. 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 REPLICATION (Prevent outbound replication processing on a server)

Use this command to prevent a source replication server from starting new replication processes.

The use of this command does not stop running replication processes. Running replication processes continue until they complete or until they end without completing. Use this command and the **ENABLE REPLICATION** command to control replication processing.

Issue this command on the server that acts as a source for replicated data.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—DISAbLe REPLicAtion—◀◀

### Parameters

None.

### Example: Disable replication processing

Disable replication processing on a source replication server.

```
disable replication
```

### Related commands

Table 153. Commands related to **DISABLE REPLICATION**

Command	Description
CANCEL REPLICATION	Cancels node replication processes.
DISABLE SESSIONS	Prevents new sessions from accessing Tivoli Storage Manager but permits existing sessions to continue.
ENABLE REPLICATION	Allows outbound replication processing on a server.
ENABLE SESSIONS	Resumes server activity following the <b>DISABLE</b> command or the <b>ACCEPT DATE</b> command.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the <b>SET</b> commands.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.

## DISABLE SESSIONS (Prevent new sessions from accessing Tivoli Storage Manager)

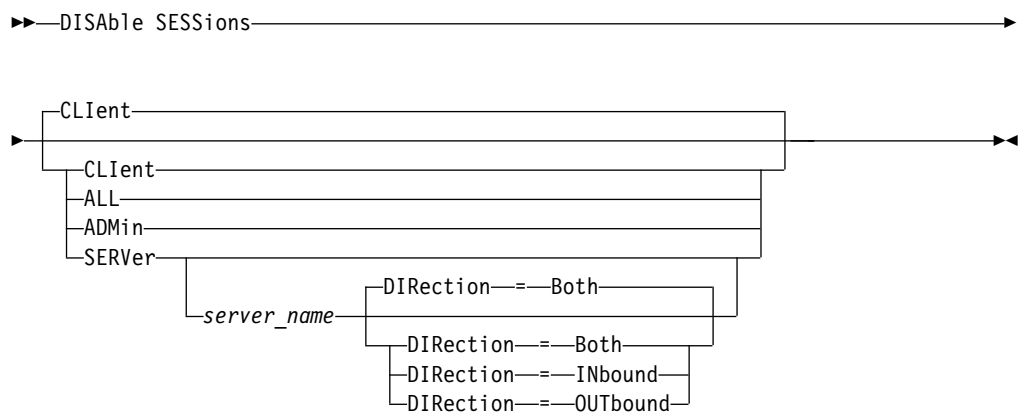
Use this command to prevent new sessions from accessing Tivoli Storage Manager. Active sessions will complete. For a particular server, you can specify whether to disable inbound sessions, outbound sessions, or both.

Server processes, such as migration and reclamation, are not affected when you issue the **DISABLE SESSIONS** 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`. You can specify one of the following values:

#### **CLient**

Disables only backup and archive client sessions.

#### **ALL**

Disables all session types.

#### **ADMin**

Disables only administrative sessions.

#### **SERVer**

Disables only server-to-server sessions. Only the following types of sessions are disabled:

- Server-to-server event logging
- Enterprise management
- Server registration
- LAN-free: storage agent - server
- Virtual volumes
- Node replication

You can also specify whether to disable inbound sessions, outbound sessions, or both for a particular server.

*server\_name*

Specifies the name of a server whose sessions you want to disable. This parameter is optional. If you do not specify this parameter, new sessions with other servers do not start. Running sessions are not canceled.

**DIRection**

Specifies whether to disable inbound sessions, outbound sessions, or both. This parameter is optional. The default is BOTH. The following values are possible:

**Both**

Specifies that inbound sessions from the specified server and outbound sessions to the specified server are disabled.

**INbound**

Specifies that only inbound sessions from the specified server are disabled.

**OUTbound**

Specifies that only outbound sessions to the specified server are disabled.

**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
```

**Example: Disable outbound sessions to a server**

Disable outbound sessions to a server named REPLSRV.

```
disable sessions server replsrv direction=outbound
```

**Related commands**

*Table 154. Commands related to DISABLE SESSIONS*

Command	Description
CANCEL SESSION	Cancels active sessions with the server.
DISABLE REPLICATION	Prevents outbound replication processing on a 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 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.

- “DISMOUNT VOLUME (Dismount a volume by volume name)” on page 466

## 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 155. 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 156. 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.

- “ENABLE EVENTS (Enable server or client events for logging)” on page 469
- “ENABLE REPLICATION (Allow outbound replication processing on a server)” on page 472
- “ENABLE SESSIONS (Resume user activity on the server)” on page 473

## 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.

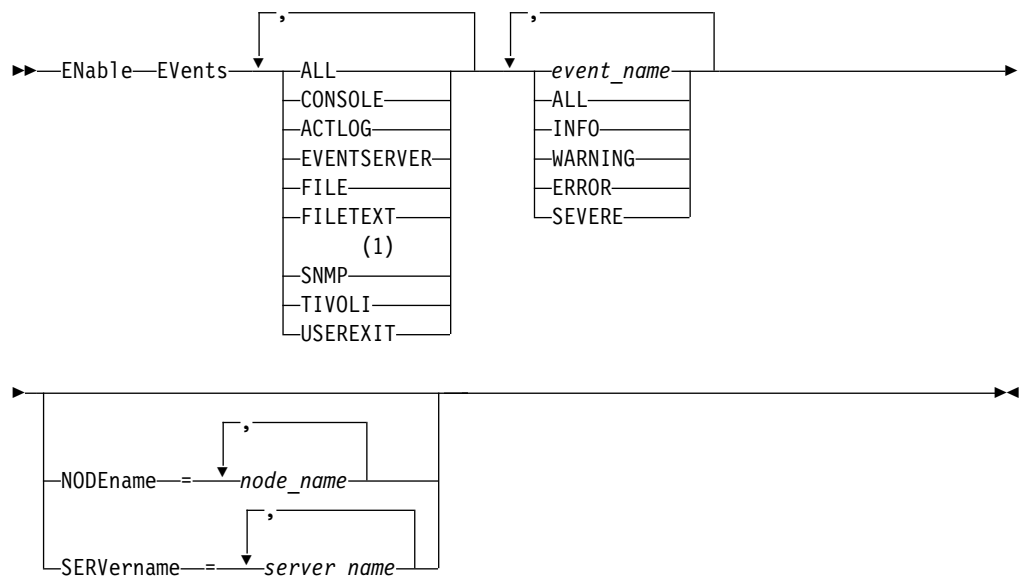
**Restriction:** Certain events, such as some messages issued during server start-up and shutdown, automatically go to the console. They do not go to other receivers even if they are enabled.

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



### Notes:

1 This parameter is only available for the Linux operating system.

### 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.

**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 157. Commands related to ENABLE EVENTS*

<b>Command</b>	<b>Description</b>
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 REPLICATION (Allow outbound replication processing on a server)

Use this command to allow a source replication server to begin normal replication processing after a database restore. You can also use this command to resume replication processing after issuing the **DISABLE REPLICATION** command.

**Attention:** Before enabling replication after a database restore, determine whether copies of data that are on the target server are needed. If they are, you must synchronize client node data by replicating the data from the target replication server to the source replication server. The replication process replaces the data on the source server that was lost because of the database restore.

Issue this command on the server that acts as a source for replicated data.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—ENable REPLication—►►

### Parameters

None.

### Example: Allow replication processing

Allow replication processing on a source replication server.

```
enable replication
```

### Related commands

Table 158. Commands related to ENABLE REPLICATION

Command	Description
DISABLE REPLICATION	Prevents outbound replication processing on a server.
DISABLE SESSIONS	Prevents new sessions from accessing Tivoli Storage Manager but permits existing sessions to continue.
ENABLE SESSIONS	Resumes server activity following the DISABLE command or the ACCEPT DATE command.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.

## ENABLE SESSIONS (Resume user activity on the server)

Use this command after issuing the **DISABLE SESSIONS** command to start new sessions that can access a server. For a particular server, you can specify whether to enable inbound sessions, outbound sessions, or both.

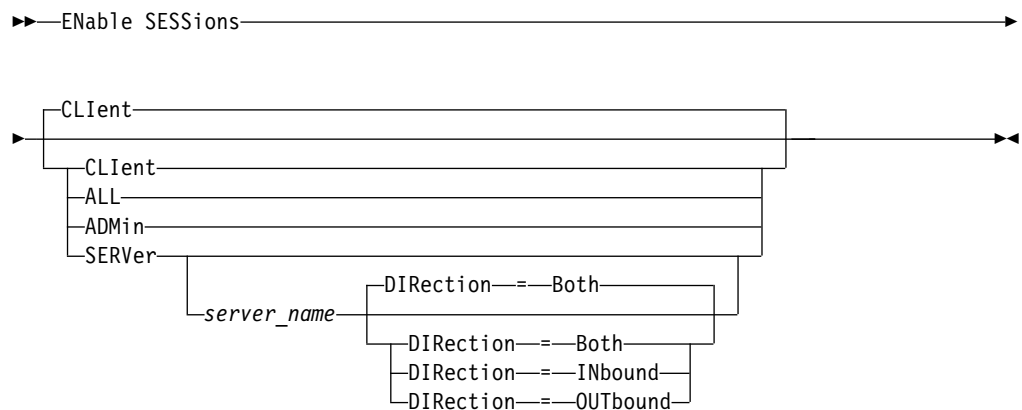
The processing of this command does not affect system processes, such as migration and reclamation.

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. You can specify one of the following values:

#### CLient

Enables only backup and archive client sessions.

#### ALL

Enables all session types.

#### ADMin

Enables only administrative sessions.

#### SERVer

Enables only server-to-server sessions. You can also specify whether to enable inbound sessions, outbound sessions, or both for a particular server.

#### *server\_name*

Specifies the name of a particular server whose sessions you want to enable. This parameter is optional. If you do not specify this parameter, new sessions with all other servers are enabled.

#### DIRection

Specifies whether to enable inbound sessions, outbound sessions, or both. This parameter is optional. The default is BOTH. The following values are possible:

**Both**

Specifies that inbound sessions from the specified server and outbound sessions to the specified server are enabled.

**INbound**

Specifies that only inbound sessions to the specified server are enabled.

**OUTbound**

Specifies that only outbound sessions from the specified server are enabled.

**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
```

**Example: Enable outbound sessions to a server**

Enable outbound sessions to a server named REPLSRV.

```
enable sessions server replsrv direction=outbound
```

**Related commands**

*Table 159. Commands related to ENABLE SESSIONS*

<b>Command</b>	<b>Description</b>
ACCEPT DATE	Accepts the current date on the server.
CANCEL SESSION	Cancels active sessions with the server.
ENABLE REPLICATION	Allows outbound replication processing on a server.
DISABLE SESSIONS	Prevents new sessions from accessing Tivoli Storage Manager but permits existing sessions to continue.
QUERY SESSION	Displays information about all active administrator and client sessions with Tivoli Storage Manager.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.



---

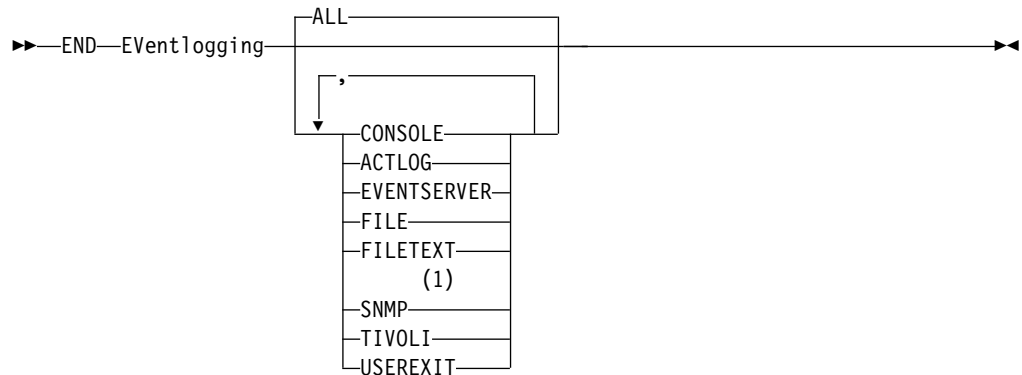
## END EVENTLOGGING (Stop logging events)

Use this command to stop logging events to an active receiver.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Notes:

- 1 This parameter is only available for the Linux operating system.

### 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 160. 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. Removal is based on policy specifications 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 RPFIL and RPFNSNAPSHOT. 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 a maximum of 40 threads. If an expiration process is running, you cannot start another process.

You can set up automatic expiration processing with the EXPINTERVAL server option. If you set the EXPINTERVAL option to 0, the server does not run expiration automatically, and you must issue the **EXPIRE INVENTORY** command to start expiration processing.

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

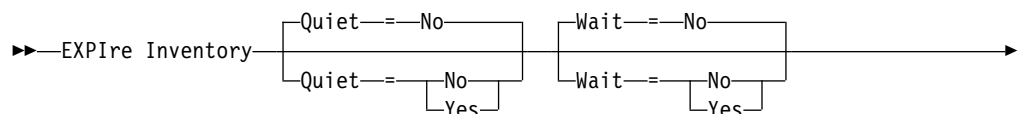
If this command is applied to a WORM volume, the volume returns to being a scratch volume if it has remaining space in which data can be written. Data on WORM volumes, including deleted and expired data, cannot be overwritten. Therefore, data can be written only in space that does not contain current, 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.

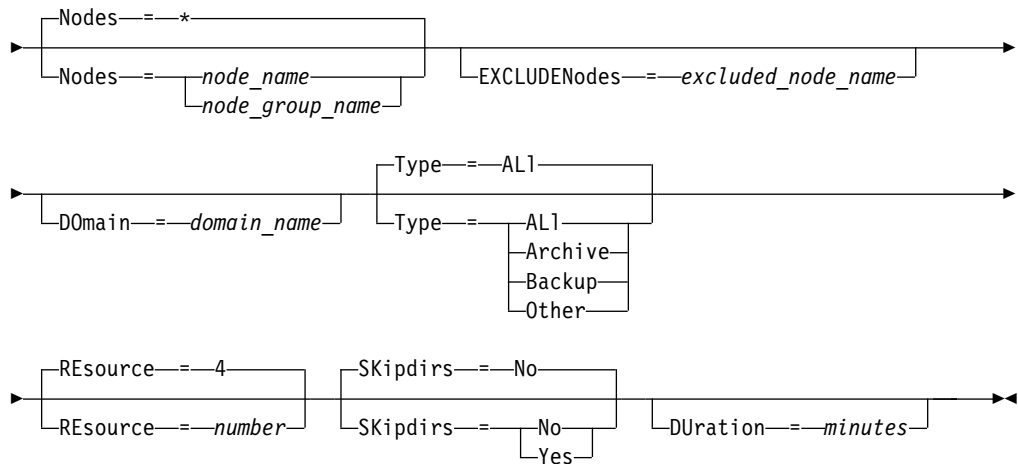
Run the **EXPIRE INVENTORY** command to delete files from server storage if they were not deleted when you used client delete operations.

### 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 was used to expire the files.

You can also specify the EXPQUIET option in the server options file to automatically determine whether expiration processing is run 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 you continue 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.

### SKipdirs

Specifies whether the server skips directory type objects during the expiration processing. This parameter is optional. The default is NO. Possible values are:

**No** Specifies that the server expires files and directories that are based on the appropriate policy criteria.

**Yes**

Specifies that the server skips directory type backup and archive objects during expiration processing, even if the directories are eligible for expiration. By specifying YES, you prevent deletion of directories, and expiration processing can occur more quickly.

**Attention:** Do not use this option all of the time. With Tivoli Storage Manager Version 6.0 and later, you can run multiple threads (resources) for an expiration process. Also, if you specify YES often, the database grows as the directory objects accumulate, and the time that is spent for expiration increases. Run SKIPDIRS=NO periodically to expire the directories and reduce the size of the database.

**Nodes**

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 names with commas and no intervening spaces. Node names can contain wildcard characters, but node group names cannot. This parameter is optional.

You can specify **NODES**, **EXCLUDENODES**, **DOMAIN**, or any combination. If you specify more than one of these parameters, only those nodes that match the criteria for both **NODES** and **DOMAIN** and does not match the criteria for **EXCLUDENODES** command options are processed. If you do not specify **NODES**, **EXCLUDENODES**, or **DOMAIN** with a value, data for all nodes is processed.

**EXCLUDENodes**

Specifies the name of the client nodes or node groups whose data is not to be processed. 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. This parameter is optional.

You can specify **NODES**, **EXCLUDENODES**, **DOMAIN**, or any combination. If you specify more than one of these parameters, only those nodes that match the criteria for both **NODES** and **DOMAIN** and does not match the criteria for **EXCLUDENODES** command options are processed. If you do not specify **NODES**, **EXCLUDENODES**, or **DOMAIN** with a value, data for all nodes is processed.

**Domain**

Specifies that only data for client nodes that are assigned to the specified domain is to be processed. This parameter is optional. You can specify **NODES**, **EXCLUDENODES**, **DOMAIN**, or any combination. If you specify more than one of these parameters, only those nodes that match the criteria for both **NODES** and **DOMAIN** and does not match the criteria for **EXCLUDENODES** command options are processed. If you do not specify **NODES**, **EXCLUDENODES**, or **DOMAIN** with a value, data for all nodes is processed.

**Type**

Specifies the type of data to be processed. This parameter is optional. The default value is **ALL**. Possible values are:

**ALL**

Process all types of data that is 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 value in the range 1 - 40. This parameter is optional. The default is four.

Expiration runs as a single process, although the resources represent parallel work by the server within the single expiration process. Archive data for a node runs only on a single resource, but backup data can be spread across resources on a file space level. For example, if you specify NODE=X,Y,Z each with three file spaces and RESOURCE=5, then expiration processing for the three X, Y, and Z client nodes runs in parallel. At least one resource processes each node, and at least one node uses multiple resources for processing backup data across the multiple file spaces.

**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. Specify a value in the range 1 - 2880. 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. Allow the server to run expiration processing until completed.

```
expire inventory nodes=charlie,robbie resource=2 type=backup
```

**Example: Run inventory expiration processing for all client nodes except two nodes**

Run inventory expiration processing for all client nodes except two nodes, CHARLIE and ROBBIE. Allow the server to run expiration processing until completed.

```
expire inventory excludenodes=charlie,robbie
```

**Example: Run inventory expiration processing for all client nodes in a domain except one node**

Run inventory expiration processing for all client nodes in a domain except one node, ROBBIE. Allow the server to run expiration processing until completed.

```
expire inventory domain=standard excludenodes=robbie
```

## Related commands

*Table 161. Commands related to EXPIRE INVENTORY*

<b>Command</b>	<b>Description</b>
AUDIT LICENSES	Verifies compliance with defined licenses.
CANCEL EXPIRATION	Cancels inventory expiration processing.
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.

**Important:** For commands that export administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **EXPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

- “EXPORT ADMIN (Export administrator information)” on page 483
- “EXPORT NODE (Export client node information)” on page 490
- “EXPORT POLICY (Export policy information)” on page 512
- “EXPORT SERVER (Export server information)” on page 519



## 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:** For commands that export administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **EXPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

Tivoli Storage Manager exports administrator information such as:

- Administrator name, password, and contact information
- Administrative privilege classes that are granted to the administrator
- 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 canceled with the **CANCEL PROCESS** command. If you export information to sequential media and the background process is canceled, the sequential media that is holding the exported data is incomplete, it must not be used for importing data. If a server-to-server export background process is canceled, a partial import might result. Evaluate any imported data on the target server to determine whether you want to keep or delete the imported data. Review the import messages for details. To display information about background processes, use the **QUERY PROCESS** command.

The following restrictions apply to the export function:

- Export operations from a later version and release to an earlier version and release is not supported.
- Export operations between servers that are at the same version and release but with different fix packs might fail. For example, you cannot export from a V7.1.3 server to a V7.1.1 or earlier server.
- Exported data from a server with retention protection enabled is not protected by retention when it is imported to another server.
- Export processing excludes nodes of type network-attached storage (NAS).
- Exporting data to a Centera device class or importing data from a Centera device class is not supported. However, files that are stored in Centera storage pools can be exported and files that must be imported can be stored on a Centera storage device.

**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server

limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

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.

- “EXPORT ADMIN (Export administrator definitions to sequential media)” on page 485
- “EXPORT ADMIN (Export administrator information directly to another server)” on page 488

*Table 162. Commands related to EXPORT ADMIN*

<b>Command</b>	<b>Description</b>
CANCEL PROCESS	Cancels a background server process.
EXPORT NODE	Copies client node information to external media or directly to another server.
EXPORT POLICY	Copies policy information to external media or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
IMPORT ADMIN	Restores administrative information from external media.
QUERY ACTLOG	Displays messages from the server activity log.
QUERY PROCESS	Displays information about background processes.

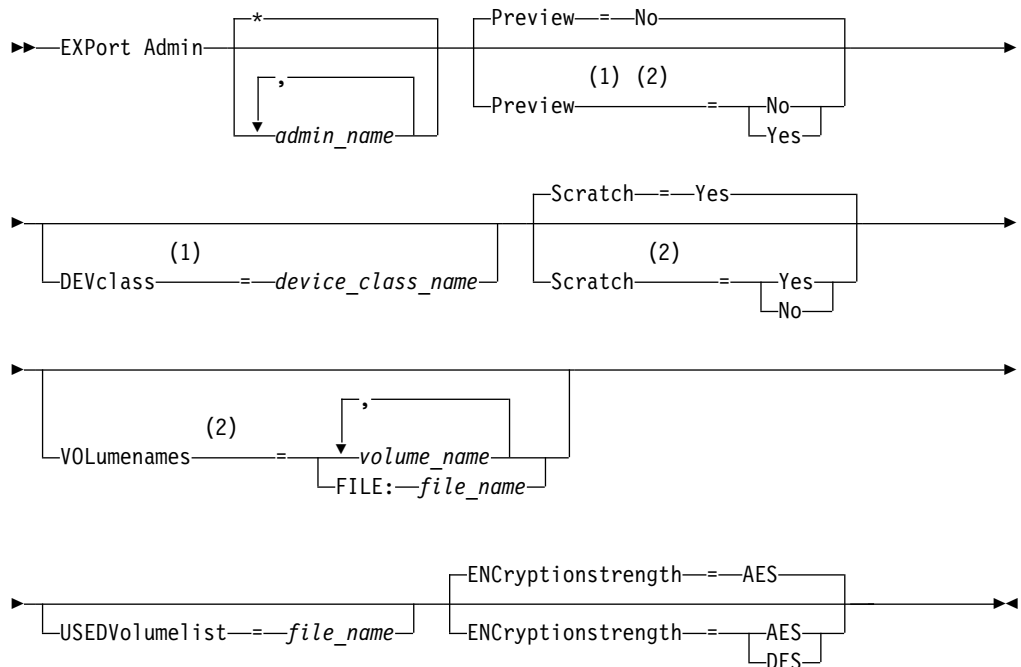
## EXPORT ADMIN (Export administrator definitions to sequential media)

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, and determine how many volumes are required. The following parameter values are supported:

- 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 is previewed but not completed. 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.

This parameter is optional. The default value is NO.

**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 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.

**Scratch**

Specifies whether scratch volumes can be used. The default value is YES. You can specify one of the following values:

**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.

**VOLumentnames**

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.

You can specify one of the following values:

*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:  /imdata/mt1.
SERVER	1-250 alphanumeric characters.

**USEDVolumeList**

Specifies the file where a list of volumes used in the export operation are stored. This parameter is optional.

This file can be used in the import operation. This file contains 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. You can specify one of the following values:

**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 that is assigned to the MENU1 device class. The number and types of objects that are 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 that is assigned to the MENU1 device class. Issue the command:

```
export admin devclass=menu1 volumenames=file:tapevol
```

The number and types of objects that are exported are reported to the system console and in the activity log.

## EXPORT ADMIN (Export administrator information directly to another server)

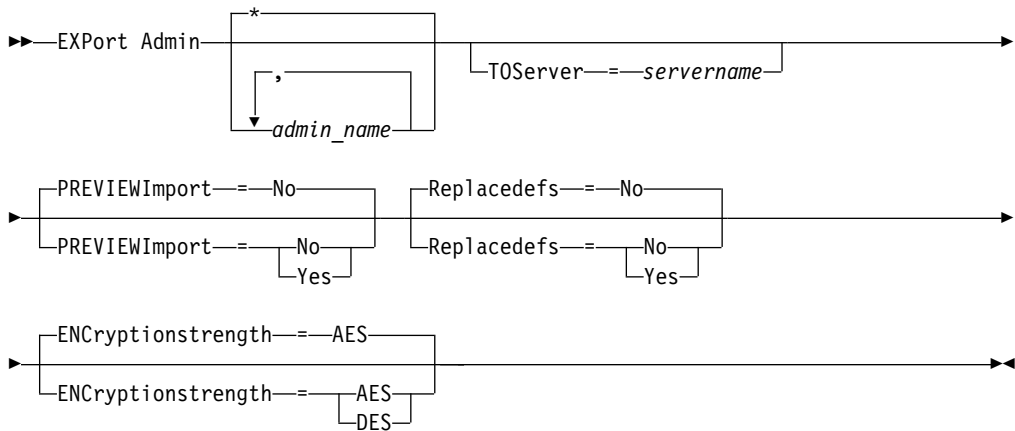
Use this command to export administrator and authority definitions directly to another server on the network. This results in an immediate import on the target server.

You can issue a **QUERY PROCESS** command from the target server to monitor the progress of the import operation. See “EXPORT ADMIN (Export administrator information)” on page 483 for a list of restrictions that apply to the export function.

### 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 is 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 is 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.

**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 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. You can specify one of the following values:

**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.

**Important:** For commands that export administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **EXPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

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 that pertains to the file spaces exported.
- Archive data that is in deletion hold status (the hold status is preserved). When the archive data is imported, it remains in deletion hold.

If you use an LDAP directory server to authenticate passwords, any servers that you export to must be configured for LDAP passwords. Node data that is exported from a node that authenticates with an LDAP directory server is inaccessible if the target server is not properly configured. If your target server is not configured, exported data from an LDAP node can still be exported. But the target server must be configured to use LDAP, to access the data.

The following restrictions apply to the export function:

- Export operations from a later version and release to an earlier version and release is not supported.
- Export operations between servers that are at the same version and release but with different fix packs might fail. For example, you cannot export from a V7.1.3 server to a V7.1.1 or earlier server.
- Exported data from a server with retention protection enabled is not protected by retention when it is imported to another server.
- Export processing excludes nodes of type network-attached storage (NAS).
- Exporting data to a Centera device class or importing data from a Centera device class is not supported. However, files that are stored in Centera storage pools can be exported and files that must be imported can be stored on a Centera storage device.
- The **EXPORT NODE** and **EXPORT SERVER** commands do not export data from a shred pool unless you explicitly allow it by setting the **ALLOWSHREDDABLE** parameter to the YES value. If this value is specified, and the exported data includes data from



shred pools, that data cannot be shredded. A warning is not issued if the export operation includes data from shred pools.

- Incrementally exporting or importing the following types of client data to another Tivoli Storage Manager server is not supported:
  - VMware backups where full plus incremental backups need to be periodically, incrementally transferred to another server
  - Backups groups where full plus differential backups must be periodically, incrementally transferred to another server
  - Windows System State data that is periodically, incrementally transferred to another server

Full export or import of this data to a new file system on the target is supported by exporting the entire file space that contains the data. The export must not use the **FILEDATA=ALLACTIVE**, **FROMDATE**, **TODATE**, or **MERGEFILESACES** parameters.

Using node replication to incrementally transfer this type of client data between two servers is optimal.

**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

The **EXPORT NODE** command generates a background process that can be canceled with the **CANCEL PROCESS** command. If you are exporting node information to sequential media and the background process is canceled, the sequential media that is holding the exported data is incomplete, it must not be used to import data. If a server-to-server export background process is canceled, a partial import might result. Evaluate any imported data on the target server to determine whether you want to keep or delete the imported data. Review the import messages for details. To display information about background processes, issue the **QUERY PROCESS** command.

To display information about any running and suspended server-to-server export operations, issue the **QUERY EXPORT** command. The **QUERY EXPORT** command displays information only for exports that are, or can be, suspended. Export operations that can be suspended, and then 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.

Because of unpredictable results, do not run expiration, migration, backup, or archive when you are issuing the **EXPORT NODE** command.

For a server that has clients with support for Unicode, you can get the server to 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.

- “EXPORT NODE (Export node definitions or file data directly to another server)” on page 502
- “EXPORT NODE (Export node definitions to sequential media)” on page 493

*Table 163. Commands related to EXPORT NODE*

<b>Command</b>	<b>Description</b>
CANCEL EXPORT	Deletes a suspended export operation.
CANCEL PROCESS	Cancels a background server process.
COPY ACTIVEDATA	Copies active backup data.
EXPORT ADMIN	Copies administrative information to external media or directly to another server.
EXPORT POLICY	Copies policy information to external media or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
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.

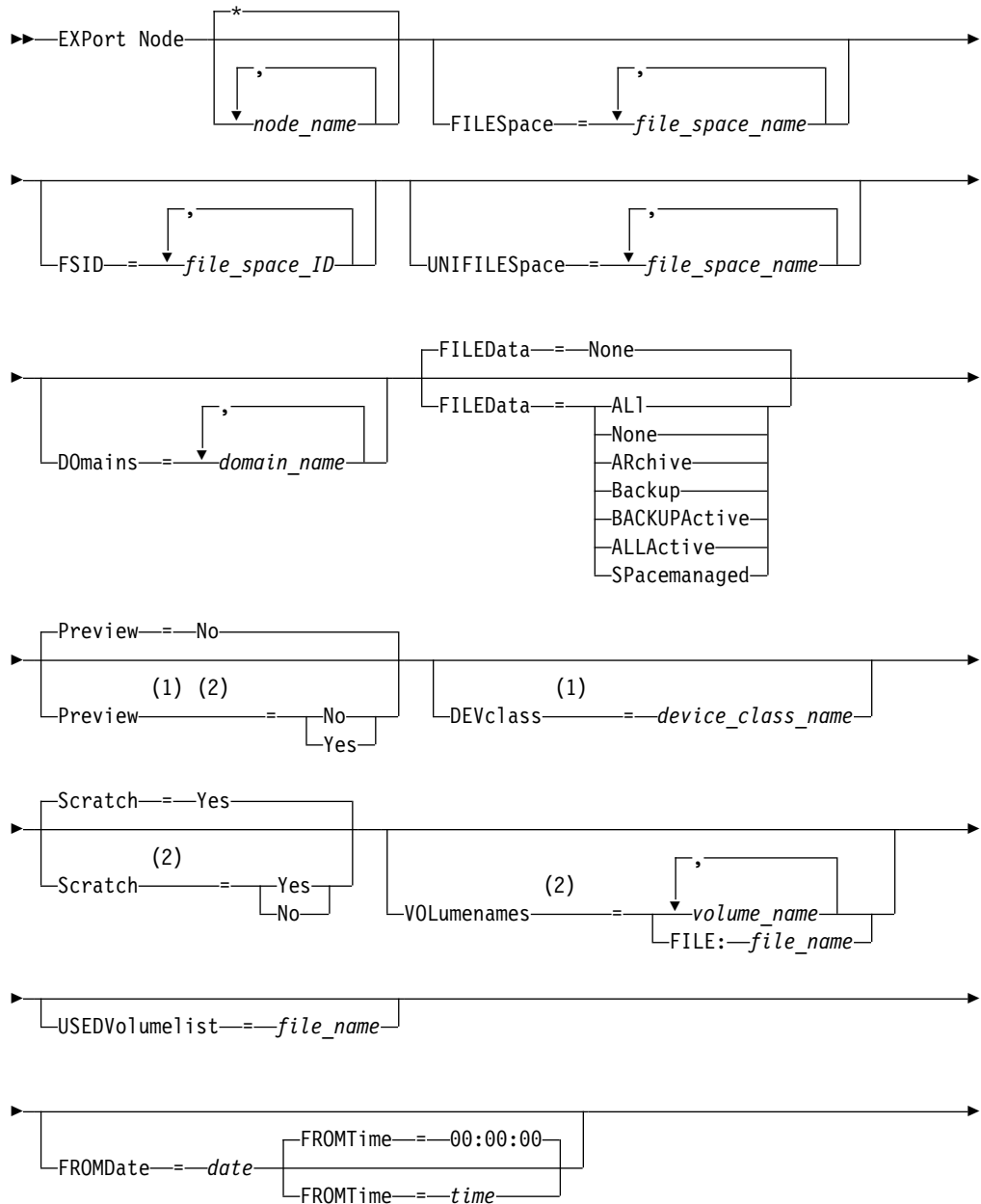
## EXPORT NODE (Export node definitions to sequential media)

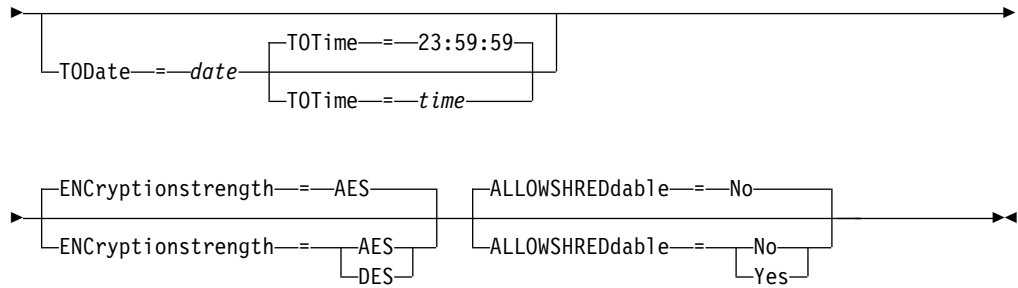
You can export node definitions or file data from a server to sequential media for later importing to another server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax





**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 are searched.

**Restriction:** If you use wildcard characters to specify a pattern for node names, the server does 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.

**Restriction:** If a file space is specified, Unicode enabled file spaces are not exported.

**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 as Unicode enabled. The server converts the names that 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. A wildcard character can be used to specify a name. This parameter is optional.

**D0mains**

Specifies the policy domains from which nodes are to 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 are to be exported for all nodes that are being exported to the server. This parameter is optional. The default value is NONE.

**Note:** If you are exporting a node that has group data, data that is not a part of the target objects might be exported. An example of group data is virtual machine data or system state backup data. For example, if FILEDATA=BACKUPACTIVE when the FROMDATE or TODATE parameters are specified, it is possible to include inactive backup data. The incremental backup processing for the data can cause extra files that do not meet the filtering criteria to be exported.

If you are exporting to sequential media: the device class that is used by the file data is determined by the device class for the storage pool. If it is the same device class that is 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. This parameter supports the following values:

**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 so that you can determine how many volumes are required. This parameter supports the following values:

**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 is previewed but not completed. 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.

This parameter is optional. The default value is NO.

**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 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.

**Scratch**

Specifies whether scratch volumes can be used. The default value is YES. You can specify one of the following values:

**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.

**VOLumentnames**

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.

You can specify one of the following values:

*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:  /imdata/mt1.
SERVER	1-250 alphanumeric characters.

**USEDVolumeList**

Specifies the file where a list of volumes used in the export operation are stored. This parameter is optional.

This file can be used in the import operation. This file contains 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 files to be exported were stored on the server. Files that were stored on the server earlier than the specified date are not exported. This parameter only applies to client file data. This parameter does not affect other information that might be exported, for example, policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

**Directory processing:** The FROMDATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. Group data on the node is, for example, virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported, so that there is a consistent image for the backup data.

Use one of the following values to specify the date:

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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

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.

When a server-to-server export operation uses a relative FROMDATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the FROMDATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted ten days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

### TODate

Specifies the latest date for files to be exported from the server. Files 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.
- If a TODATE parameter is specified without a TOTIME parameter, the server exports all objects that are inserted on or before the day specified by the TODATE parameter.
- 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.
- The TODATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

**Important:** If you have group data on the node that you are exporting, data that was backed up after the TODATE or TOTIME parameters can be exported. An example of group data is virtual machine data or system state backup data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported, so that there is a consistent image for the backup data.

Use one of the following values to specify the date:

Value	Description	Example
MM/DD/YYYY	A specific date	10/15/2006
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 -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM-days	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.



Value	Description	Example
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9 To include files that were active on the 10th day of the current month.

When a server-to-server export operation uses a relative TODATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the TODATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted 10 days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

#### 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 does not affect other information that might be exported, for example, policy. Objects that were stored on the server before the specified time and date are not exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. An example of group data on the node is virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported so that there is a consistent image for the backup data.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

Use one of the following values to specify the time:

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 <b>+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 or +02:00. 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.

Value	Description	Example
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes specified	NOW -02:00 or -02:00.  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.

Use one of the following values to specify the time:

Value	Description	Example
<i>HH:MM:SS</i>	A specific time	10:30:08
NOW+ <i>HH:MM</i> or + <i>HH:MM</i>	The current time plus hours and minutes specified.	NOW+02:00 or +02:00.  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.
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes specified.	NOW-02:00 or -02:00.  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. You can specify one of the following values:

##### 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 supports the following values:

**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 is not shredded.

This parameter is optional. The default value is NO.

**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 that is assigned to the MENU1 device class.

```
export node devclass=menu1 volumenames=tape01,tape02,tape03
```

**Example: Export client node information by 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 filesystem 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 is used by a device that is assigned to the MENU1 device class.

```
export node joe fsid=1,2 filedata=backupactive devclass=menu1 volumenames=tape01
```

**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 that is 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 excludes these nodes.

You can suspend and 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 can be restarted from the point at which the operation failed or was suspended. The export operation can be restarted later 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 that is 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.

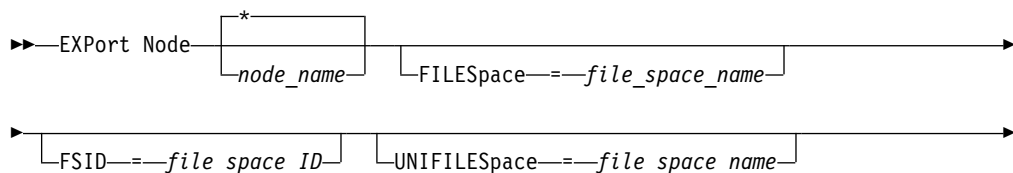
The export operation cannot be restarted if the export operation fails before transmitting the eligible node and file space definitions to the target server. You must reenter the command to begin a new export operation.

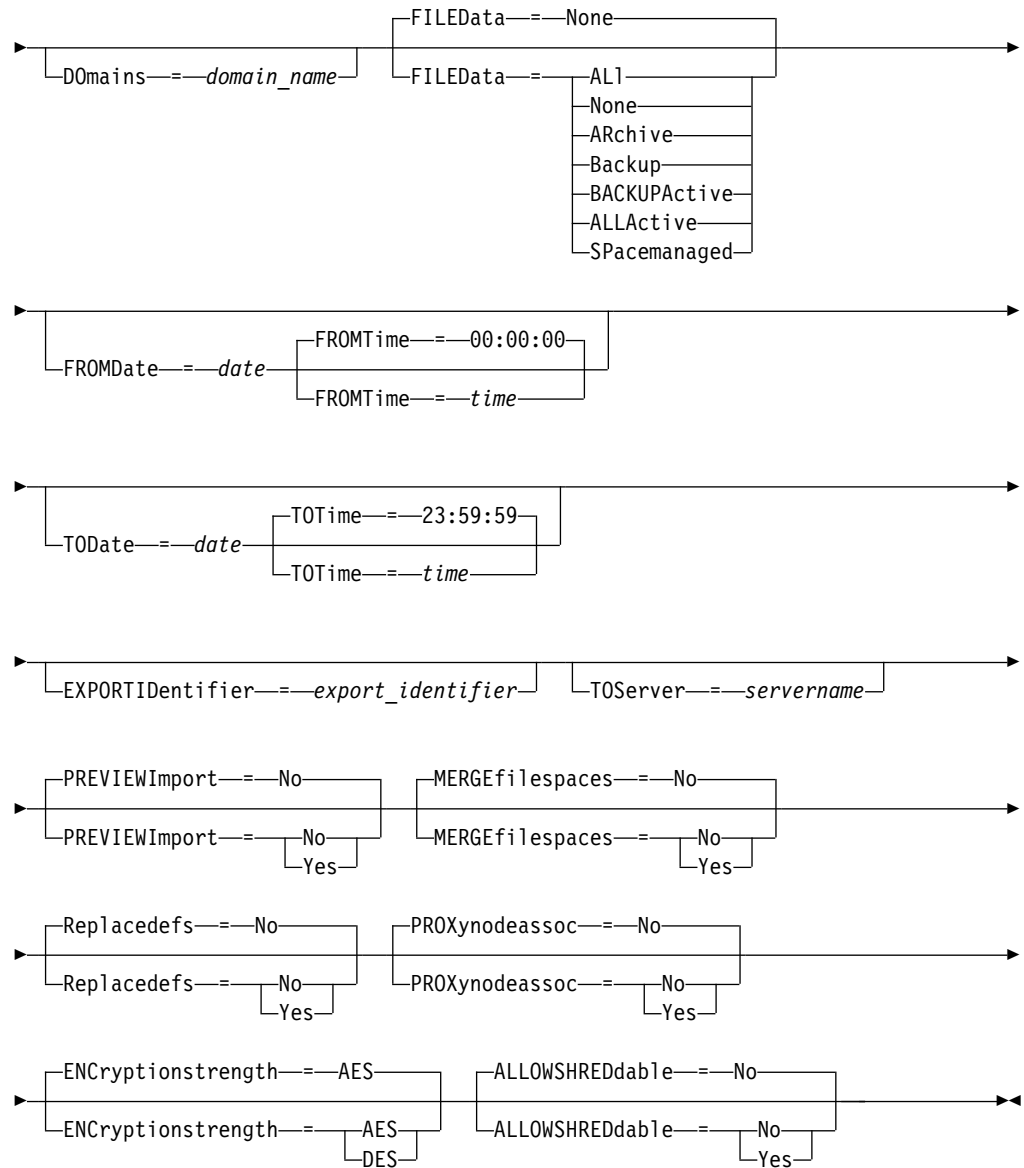
You can issue a **QUERY PROCESS** command from the target server to monitor the progress of the import operation. Issue the **QUERY EXPORT** command to list all restartable server-to-server export operations. See “EXPORT ADMIN (Export administrator information)” on page 483 for a list of restrictions that apply to the export function.

### Privilege class

To issue this command, you must have system privilege.

### Syntax





## 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 are searched.

**Restriction:** If you specify a list of node names or node patterns, the server does 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.

**Restriction:** If a file space is specified, no Unicode enabled file spaces are exported.

#### **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 that 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. A wildcard character can be used to specify a name. This parameter is optional.

#### **DOmains**

Specifies the policy domains from which nodes are 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.

**Note:** If you are exporting a node that has group data, data that is not a part of the target objects might be exported. An example of group data is virtual machine data or system state backup data. For example, if **FILEDATA=BACKUPACTIVE** when the **FROMDATE** or **TODATE** parameters are specified, it is possible to include inactive backup data. The incremental backup processing for the data can cause extra files that do not meet the filtering criteria to be exported.

If you are exporting to sequential media, the device class that is used by the file data is determined by the device class for the storage pool. If it is the same device class that is 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 as follows:

#### **ALl**

The server exports all backup versions of files, all archived files, and all files that are 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 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 files to be exported were stored on the server. Files that were stored on the server earlier than the specified date are not exported. This parameter only applies to client file data. This parameter does not affect other information that might be exported, for example, policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

**Directory processing:** The FROMDATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. Group data on the node is, for example, virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported, so that there is a consistent image for the backup data.

Use one of the following values to specify the date:

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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.

Value	Description	Example
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

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.

When a server-to-server export operation uses a relative FROMDATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the FROMDATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted ten days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

#### TODate

Specifies the latest date for files to be exported from the server. Files that are 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.
- If a TODATE parameter is specified without a TOTIME parameter, the server exports all objects that are inserted on or before the day specified by the TODATE parameter.
- 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.
- The TODATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

**Important:** If you have group data on the node that you are exporting, data that was backed up after the TODATE or TOTIME parameters can be exported. An example of group data is virtual machine data or system state backup data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported, so that there is a consistent image for the backup data.

Use one of the following values to specify the date:

Value	Description	Example
MM/DD/YYYY	A specific date	10/15/2006
TODAY	The current date	TODAY



Value	Description	Example
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

When a server-to-server export operation uses a relative TODATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the TODATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted 10 days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

#### 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 does not affect other information that might be exported, for example, policy. Objects that were stored on the server before the specified time and date are not exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. An example of group data on the node is virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported so that there is a consistent image for the backup data.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

Use one of the following values to specify the time:

Value	Description	Example
HH:MM:SS	A specific time	10:30:08
NOW	The current time	NOW

Value	Description	Example
<b>NOW+HH:MM</b> or <b>+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 or +02:00.  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> or <b>-HH:MM</b>	The current time minus hours and minutes specified	NOW -02:00 or -02:00.  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.

Use one of the following values to specify the time:

Value	Description	Example
<b>HH:MM:SS</b>	A specific time	10:30:08
<b>NOW+HH:MM</b> or <b>+HH:MM</b>	The current time plus hours and minutes specified.	NOW+02:00 or +02:00.  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> or <b>-HH:MM</b>	The current time minus hours and minutes specified.	NOW-02:00 or -02:00.  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.

### T0Server

Specifies the name of a server to which the export data is 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 is 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.

#### **MERGEfilespaces**

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 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 exists.

#### **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 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`. You can specify one of the following values:

##### **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 is not 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 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. When a policy is exported by using the **EXPORT POLICY** command, the active data pool information in the domain is not 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 canceled with the **CANCEL PROCESS** command. If you export policy information to sequential media and the background process is canceled, the sequential media that is holding the exported data is incomplete and must not be used to import data. If a server-to-server export background process is canceled, a partial import might result. Evaluate any imported data on the target server to determine whether you want to keep or delete the imported data. Review the import messages for details. To display information about background processes, use the **QUERY PROCESS** command.

The following restrictions apply to the export function:

- Export operations from a later version and release to an earlier version and release is not supported.
- Export operations between servers that are at the same version and release but with different fix packs might fail. For example, you cannot export from a V7.1.3 server to a V7.1.1 or earlier server.
- Exported data from a server with retention protection enabled is not protected by retention when it is imported to another server.
- Export processing excludes nodes of type network-attached storage (NAS).
- Exporting data to a Centera device class or importing data from a Centera device class is not supported. However, files that are stored in Centera storage pools can be exported and files that must be imported can be stored on a Centera storage device.

**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

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.

- “EXPORT POLICY (Export a policy directly to another server)” on page 517
- “EXPORT POLICY (Export policy information to sequential media)” on page 514

*Table 164. Commands related to EXPORT POLICY*

<b>Command</b>	<b>Description</b>
CANCEL PROCESS	Cancels a background server process.
EXPORT ADMIN	Copies administrative information to external media or directly to another server.
EXPORT NODE	Copies client node information to external media or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
IMPORT POLICY	Restores policy information from external media.
QUERY ACTLOG	Displays messages from the server activity log.
QUERY PROCESS	Displays information about background processes.

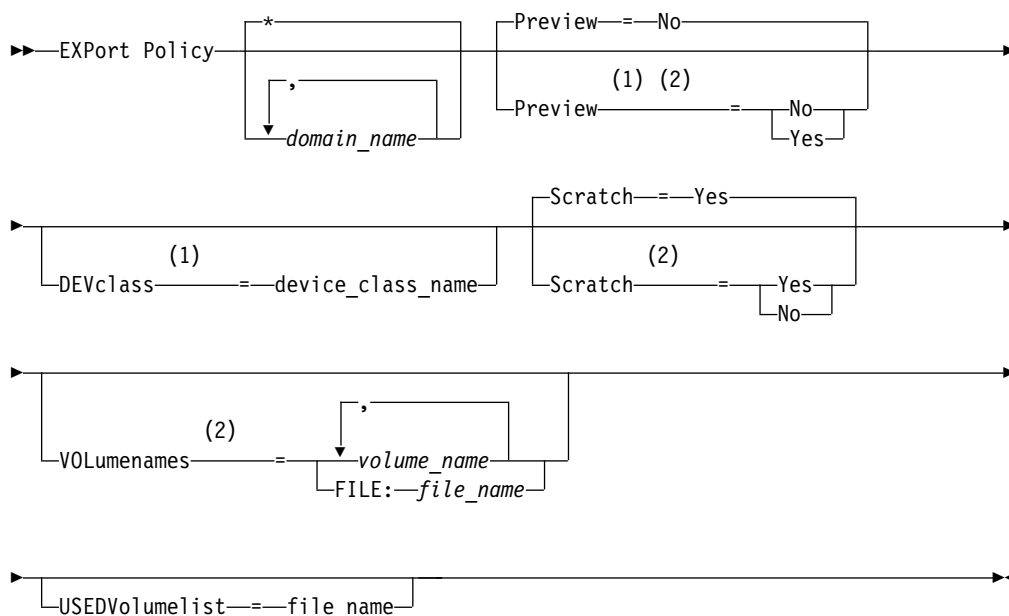
## EXPORT POLICY (Export policy information to sequential media)

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 so that you can determine how many volumes are required. This parameter supports the following values:

**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 is previewed but not completed. 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.

This parameter is optional. The default value is NO.

### **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 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.

### **Scratch**

Specifies whether scratch volumes can be used. The default value is YES. You can specify one of the following values:

#### **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.

### **VOLumentnames**

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.

You can specify one of the following values:

*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:

<b>For this device</b>	<b>Specify</b>
Tape	1-6 alphanumeric characters.
FILE	Any fully qualified file name string. For example:  /imdata/mt1.
SERVER	1-250 alphanumeric characters.

### **USEDVolumelist**

Specifies the file where a list of volumes used in the export operation are stored. This parameter is optional.

This file can be used in the import operation. This file contains 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 that is 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 that is 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)

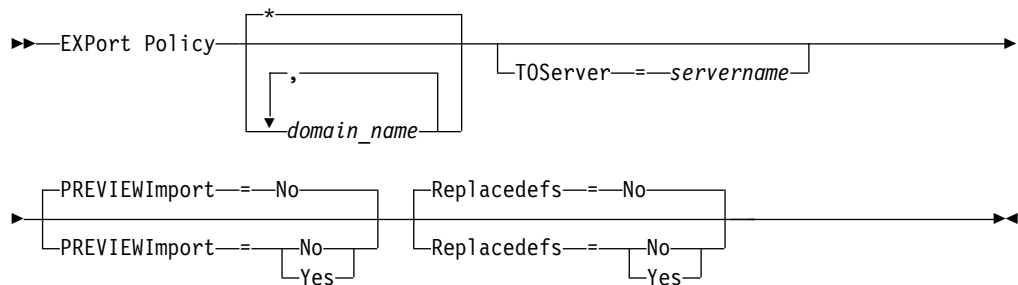
Use this command to export policy information directly to another server on the network. This results in an immediate import on the target server.

To monitor the progress of the import operation, you can issue a **QUERY PROCESS** command from the target server. See “EXPORT ADMIN (Export administrator information)” on page 483 for a list of restrictions that apply to the export function.

### 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.

#### **TOServer**

Specifies the name of a server to which the export data is 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 is 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.

**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 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:** For commands that import administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **IMPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

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 that pertains to the file spaces exported
- Backed-up, archived, and files that were migrated by a Tivoli Storage Manager for Space Management client

This command generates a background process that can be canceled by the **CANCEL PROCESS** command. If you export server information to sequential media, and the background process is canceled, 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 canceled, a partial import might result. Evaluate any imported data on the target server to determine whether 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 which indicates the size and the success or failure of the export operation.

The following restrictions apply to the export function:

- Export operations from a later version and release to an earlier version and release is not supported.

- Export operations between servers that are at the same version and release but with different fix packs might fail. For example, you cannot export from a V7.1.3 server to a V7.1.1 or earlier server.
- Exported data from a server with retention protection enabled is not protected by retention when it is imported to another server.
- Export processing excludes nodes of type network-attached storage (NAS).
- Exporting data to a Centera device class or importing data from a Centera device class is not supported. However, files that are stored in Centera storage pools can be exported and files that must be imported can be stored on a Centera storage device.
- The **EXPORT NODE** and **EXPORT SERVER** commands do not export data from a shred pool unless you explicitly allow it by setting the **ALLOWSHREDDABLE** parameter to the YES value. If this value is specified, and the exported data includes data from shred pools, that data cannot be shredded. A warning is not issued if the export operation includes data from shred pools.
- Incrementally exporting or importing the following types of client data to another Tivoli Storage Manager server is not supported:
  - VMware backups where full plus incremental backups need to be periodically, incrementally transferred to another server
  - Backups groups where full plus differential backups must be periodically, incrementally transferred to another server
  - Windows System State data that is periodically, incrementally transferred to another server

Full export or import of this data to a new file system on the target is supported by exporting the entire file space that contains the data. The export must not use the **FILEDATA=ALLACTIVE**, **FROMDATE**, **TODATE**, or **MERGEFILESACES** parameters.

Using node replication to incrementally transfer this type of client data between two servers is optimal.

**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

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 form are defined separately.

- “EXPORT SERVER (Export a server to sequential media)” on page 522
- “EXPORT SERVER (Export server control information and client file data to another server)” on page 530

*Table 165. Commands related to EXPORT SERVER*

Command	Description
CANCEL EXPORT	Deletes a suspended export operation.
CANCEL PROCESS	Cancels a background server process.
COPY ACTIVE DATA	Copies active backup data.

Table 165. Commands related to **EXPORT SERVER** (continued)

<b>Command</b>	<b>Description</b>
EXPORT ADMIN	Copies administrative information to external media or directly to another server.
EXPORT NODE	Copies client node information to external media or directly to another server.
EXPORT POLICY	Copies policy information to external media or directly to another server.
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.

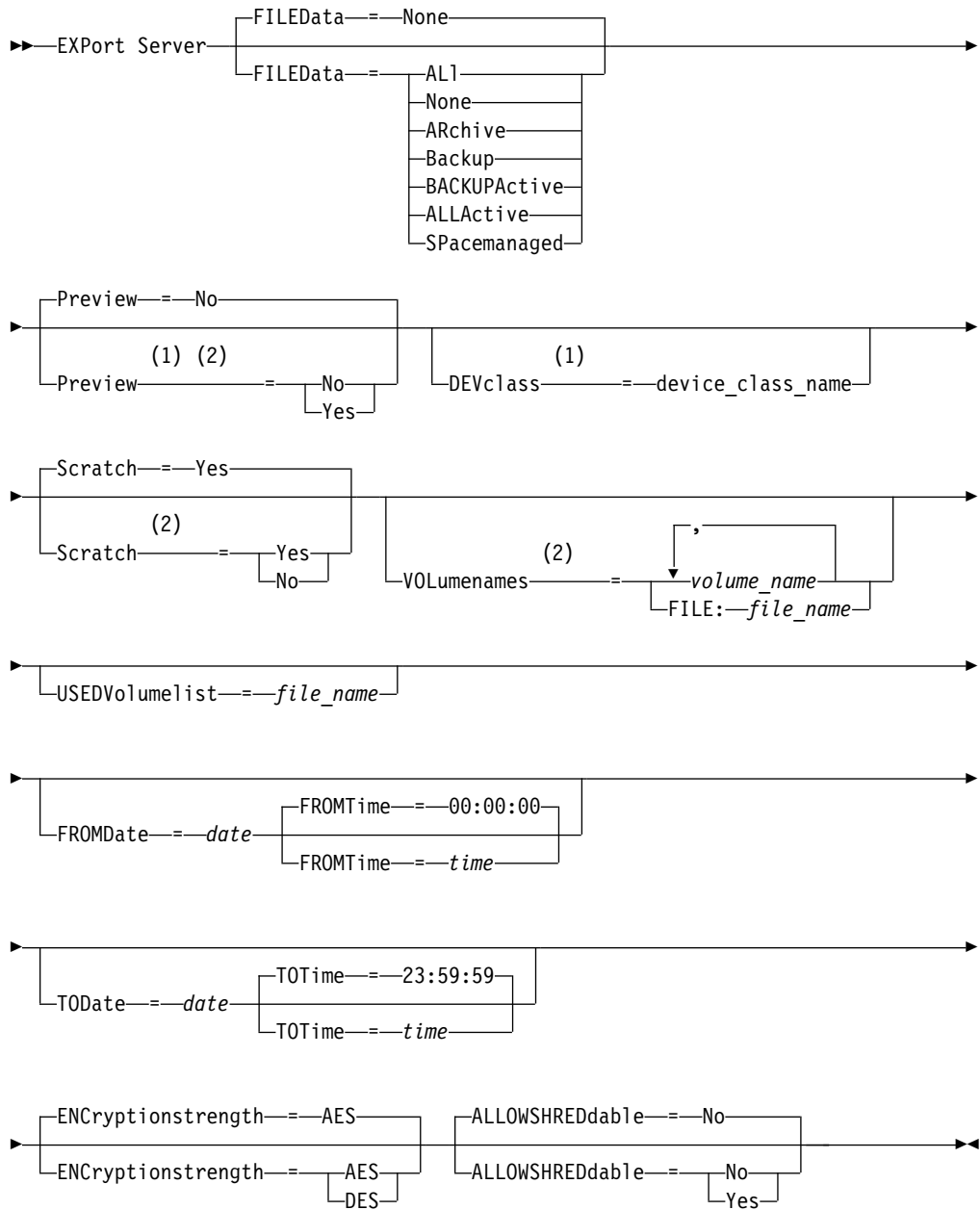
## EXPORT SERVER (Export a server to sequential media)

You can export all or part of the server control information and client file data from a server to sequential media so that this information can be imported 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 are exported for all nodes that are 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 that is 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 supports the following values:

**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 is previewed but not completed. 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.

This parameter is optional. The default value is NO.

### 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 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.

### Scratch

Specifies whether scratch volumes can be used. The default value is YES. You can specify one of the following values:

#### 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.

### VOLumentnames

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.

You can specify one of the following values:

#### *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:  /imdata/mt1.
SERVER	1-250 alphanumeric characters.

### USEDVolumelist

Specifies the file where a list of volumes used in the export operation are stored. This parameter is optional.

This file can be used in the import operation. This file contains 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 files to be exported were stored on the server. Files that were stored on the server earlier than the specified date are not exported. This parameter only applies to client file data. This parameter does not affect other information that might be exported, for example, policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

**Directory processing:** The FROMDATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. Group data on the node is, for example, virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported, so that there is a consistent image for the backup data.

Use one of the following values to specify the date:

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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

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.

When a server-to-server export operation uses a relative FROMDATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the FROMDATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted ten days later (07/14/2009), the date that is used for selecting files is still

07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

### TODate

Specifies the latest date for files to be exported from the server. Files 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.
- 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.
- 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.
- The TODATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

Use one of the following values to specify the date:

Value	Description	Example
<i>MM/DD/YYYY</i>	A specific date	10/15/2006
TODAY	The current date	TODAY
TODAY- <i>days</i> or <i>-days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

When a server-to-server export operation uses a relative TODATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the TODATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted ten days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

### 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 does not affect other information that might be exported, for example, policy. Objects that were stored on the server before the specified time and date are not exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. An example of group data on the node is virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported so that there is a consistent image for the backup data.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

Use one of the following values to specify the time:

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 <b>+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 or +02:00.  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> or <b>-HH:MM</b>	The current time minus hours and minutes specified	NOW -02:00 or -02:00.  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.

Use one of the following values to specify the time:

Value	Description	Example
<i>HH:MM:SS</i>	A specific time	10:30:08

Value	Description	Example
NOW+HH:MM or+HH:MM	The current time plus hours and minutes specified.	NOW+02:00 <b>or</b> +02:00.  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.
NOW-HH:MM or-HH:MM	The current time minus hours and minutes specified.	NOW-02:00 <b>or</b> -02:00.  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. You can specify one of the following values:

#### 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 is not 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 that is 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 that is 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

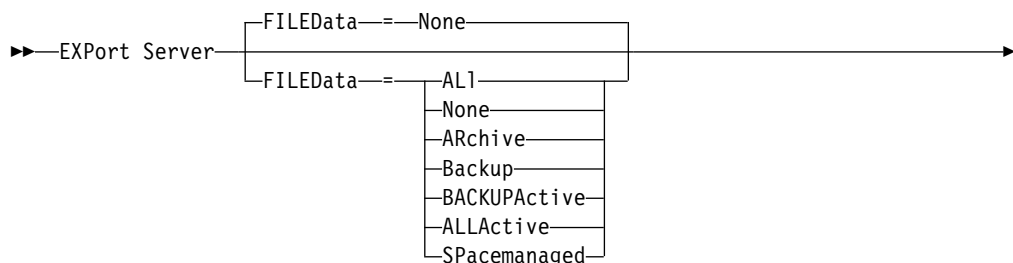
The export operation cannot be restarted if the export operation fails prior to transmitting the eligible node and filespace definitions to the target server. You must reenter the command to begin a new export operation.

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. See "EXPORT ADMIN (Export administrator information)" on page 483 for a list of restrictions that apply to the export function.

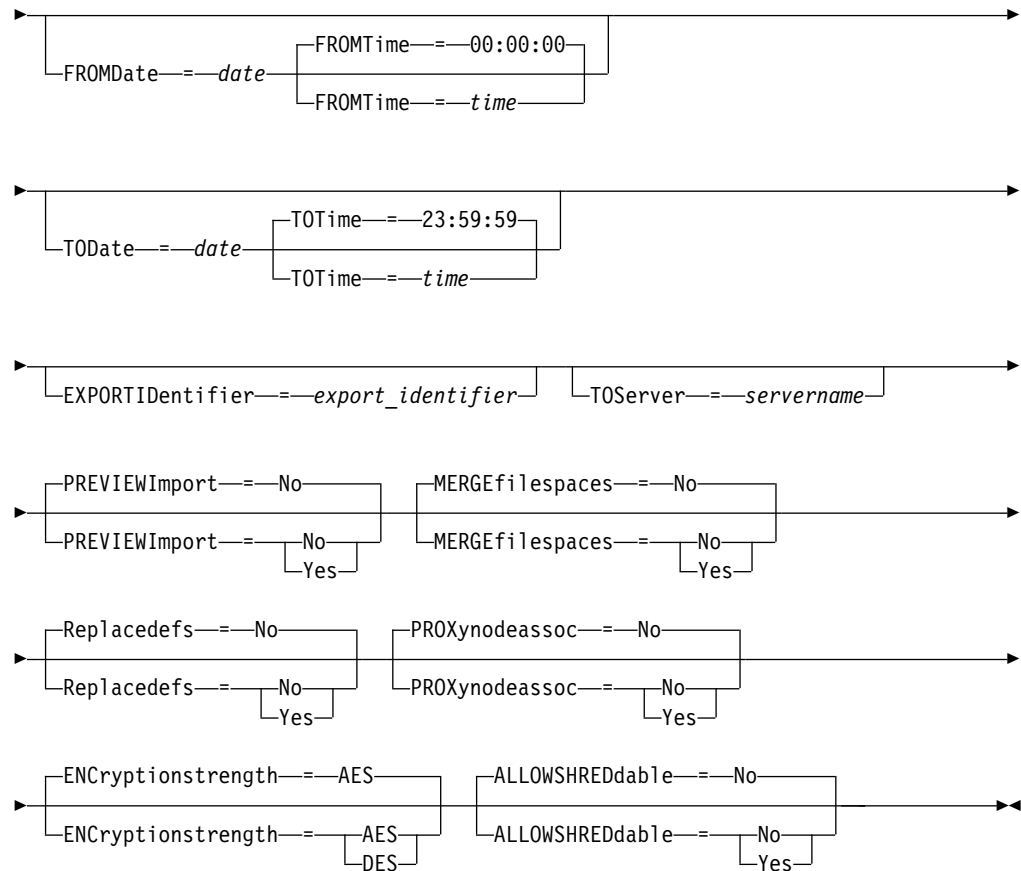
### Privilege class

To issue this command, you must have system privilege.

### Syntax







## 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 files to be exported were stored on the server. Files that were stored on the server earlier than the specified date are not exported. This parameter only applies to client file data. This parameter does not affect other information that might be exported, for example, policy. Tivoli Storage Manager ignores the FROMDATE parameter when the FILEDATA parameter is set to NONE.

**Directory processing:** The FROMDATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. Group data on the node is, for example, virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported, so that there is a consistent image for the backup data.

Use one of the following values to specify the date:

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.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM

Value	Description	Example
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

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.

When a server-to-server export operation uses a relative FROMDATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the FROMDATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted ten days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

### TODate

Specifies the latest date for files to be exported from the server. Files 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.
- 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.
- 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.
- The TODATE parameter does not apply to directories. All directories in a file space are processed even if the directories were not backed up in the specified date range.

Use one of the following values to specify the date:

Value	Description	Example
MM/DD/YYYY	A specific date	10/15/2006
TODAY	The current date	TODAY
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.

Value	Description	Example
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9 To include files that were active on the 10th day of the current month.

When a server-to-server export operation uses a relative TODATE, for example, TODAY-1, and the operation is restarted at a later date, the restarted process still uses the date that was used during the original operation. For example, if a server-to-server export operation is started on 07/04/2009 and the TODATE is specified as TODAY-1, the date that is used for selecting files is 07/03/2009. If this same export operation is suspended and restarted ten days later (07/14/2009), the date that is used for selecting files is still 07/03/2009. This behavior ensures that the entire export operation uses the same cutoff date for selecting files to export.

#### 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 does not affect other information that might be exported, for example, policy. Objects that were stored on the server before the specified time and date are not exported. Tivoli Storage Manager ignores the FROMTIME parameter when the FILEDATA parameter is set to NONE.

**Important:** If you have group data on the node that you are exporting, data that was backed up before the designated FROMDATE and FROMTIME can also be exported. An example of group data on the node is virtual machine data or system state backup data. This export is a result of incremental backup processing for the data. The incremental backup processing can cause extra files that do not meet the filtering criteria to be exported so that there is a consistent image for the backup data.

The default value for this parameter when used with the FROMDATE parameter is midnight (00:00:00).

Use one of the following values to specify the time:

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. 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.

Value	Description	Example
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes specified	NOW -02:00 or -02:00.  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.

Use one of the following values to specify the time:

Value	Description	Example
<i>HH:MM:SS</i>	A specific time	10:30:08
NOW+ <i>HH:MM</i> or+ <i>HH:MM</i>	The current time plus hours and minutes specified.	NOW+02:00 or +02:00.  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.
NOW- <i>HH:MM</i> or- <i>HH:MM</i>	The current time minus hours and minutes specified.	NOW-02:00 or -02:00.  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 is 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 is 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.

**MERGEfilespaces**

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 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 exists.

**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 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. You can specify one of the following values:

**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**, **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.

When you issue the **EXTEND DBSPACE** command, directories are added to the database. With the default parameter settings, data is redistributed across all database directories, and storage space is reclaimed. This action improves parallel I/O performance and makes the new directory space available for immediate use.

If you do not want to redistribute data when you add new directories, you can specify `RECLAIMSTORAGE=NO`. If you specify `No` for this parameter, all space in existing directories is filled before new directories are used. You can redistribute data and reclaim space later, but you must complete the manual procedure for this task by using DB2 commands.

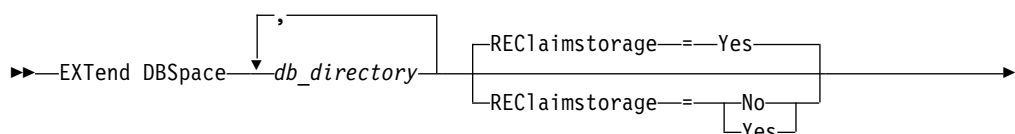
**Restriction:** Redistribution of data and reclaiming of space as part of an operation to extend database space works only with DB2 Version 9.7 or later table spaces. The table spaces are created when you format a new Tivoli Storage Manager Version 6.2 or later server. If you upgraded or restored your Tivoli Storage Manager server from V6.1, you cannot redistribute data or reclaim space. You must issue the **EXTEND DBSPACE** command with `RECLAIMSTORAGE=NO`.

**Important:** The redistribution process uses considerable system resources, so ensure that you plan ahead when you want to add space to the database. Review the following guidelines:

- Complete the process when the server is not handling a heavy workload.
- The time that is required to redistribute data and reclaim space might vary. It is affected by factors such as the file system layout, the ratio of new paths to existing storage paths, server hardware, and concurrent operations. To get a rough estimate, you can try the operation with a small Tivoli Storage Manager database on a lab system. Use your results as a reference to estimate the time that is required for the procedure.
- Do not interrupt the redistribution process. If you try to stop it, for example, by halting the process that is completing the work, you must stop and restart the DB2 server. When the server is restarted, it will go into crash recovery mode, which takes several minutes, after which the redistribution process resumes.

After an operation to extend the database space is complete, halt and restart the server to fully use the new directories. If the existing database directories are nearly full when a new directory is added, the server might encounter an out of space condition (reported in the `db2diag.log`). You can fix the out of space condition by halting and restarting the server.

### Syntax







## 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 and cannot exceed 175 characters in length. Enclose the name in quotation marks if it contains embedded blanks, an equal sign, or other special characters. If you are specifying a list of directories for database storage, the maximum length of the list can be 1400 characters.

**Tip:** Specify directories that are the same size as existing directories to ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching and distribution of the database.

### RECLAIMstorage

Specifies whether data is redistributed across newly created database directories and space is reclaimed from the old storage paths. This parameter is optional. The default value is Yes.

Unless you specify WAIT=YES, the operation is completed as a background process.

#### Yes

Specifies that data is redistributed so that new directories are available for immediate use.

**Important:** The redistribution process uses considerable system resources so ensure that you plan ahead.

After the process starts, messages are issued to inform you about the progress. You can use the **QUERY PROCESS** command to monitor the operation. To cancel the process, you can use the **CANCEL PROCESS** command, but if a data redistribution operation is in progress, it completes before the process is stopped.

**No** Specifies that data is not redistributed across database directories and storage space is not reclaimed when space is added for the database.

### Wait

Specifies whether this command is processed in the background or foreground.

**No** Specifies background processing. The default is NO.

#### Yes

Specifies foreground processing.

You cannot specify YES from the server console.

## Example: Add directories to the storage space for the database, redistribute data, and reclaim storage

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 166. Commands related to EXTEND DBSPACE*

<b>Command</b>	<b>Description</b>
DSMSERV EXTEND DBSPACE	Adds directories to increase space for use by the database.
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** commands for backup sets for a selected filespace or client node.

- “GENERATE BACKUPSET (Generate a backup set of Backup-Archive Client data)” on page 542
- “GENERATE BACKUPSETTOC (Generate a table of contents for a backup set)” on page 551

## GENERATE BACKUPSET (Generate a backup set of Backup-Archive Client data)

Use this command to generate a backup set for a Backup-Archive Client node. A *backup set* is a collection of a Backup-Archive Client's active backed up data, which is 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 be used only by a Backup-Archive Client.

**Restriction:** A backup set in “deduplication format” has that designation as a result of a **GENERATE BACKUPSET** command with at least one of the following specifications:

- Includes a node at Backup-Archive Client Version 6.1.x (at least V6.1.0 but less than V6.2.0).
- Includes a node that has one or more nodes that are authorized to act as a proxy. At least one of those proxy nodes is at Backup-Archive Client V6.1.x.

Backup sets in the deduplication format can be restored only by the V6.1.2 or later Backup-Archive Client. Backup-Archive Clients before V6.1.2 cannot restore from a backup set that is in the deduplication format.

A backup set in the “distributed deduplication format” has that designation as a result of a **GENERATE BACKUPSET** command with at least one of the following specifications:

- Includes a node at Backup-Archive Client level V6.2.0 or later.
- Includes a node that has one or more nodes that are authorized to act as a proxy. At least one of those proxy nodes is at Backup-Archive Client V6.2.0.

Backup sets in the distributed deduplication format can be restored only by the V6.2.0 or later Backup-Archive Client.

**Restriction:** You cannot generate a backup set with files that were backed up to Tivoli Storage Manager using NDMP. However, you can create a backup set with files that were backed up using NetApp SnapShot Difference.

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. The server then consolidates them onto sequential media. Currently, the backup object types that are supported for backup sets include directories and files only.

The backup-archive client node can restore its backup set from the server and from the media to which the backup set was written.

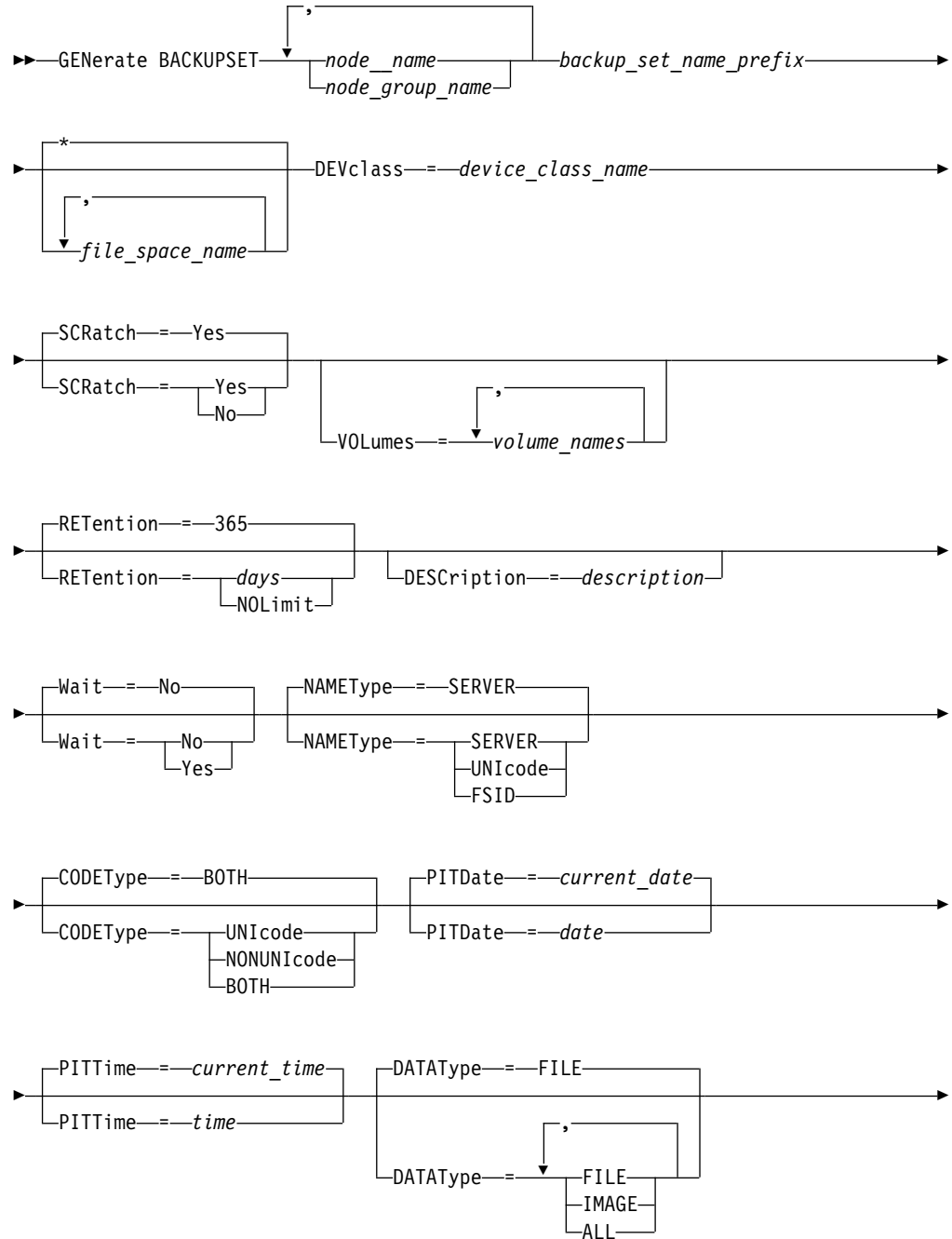
This command generates a background process that can be canceled with the **CANCEL PROCESS** command. If the background process created by this command is canceled, the media might not contain a complete backup set. You can use the **QUERY PROCESS** command to show information about the background process that is created by this command.

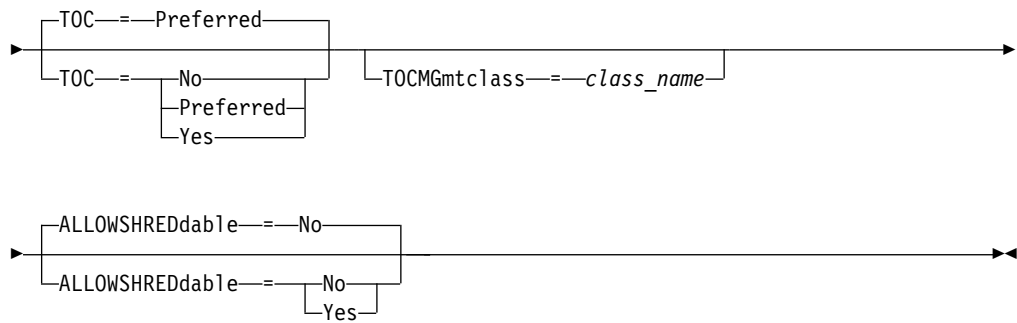
**Tip:** When Tivoli Storage Manager generates a backup set, you can improve performance if the primary storage pools containing the client data are collocated. If a primary storage pool is collocated, client node data is likely to be on fewer tape volumes than it would be if the storage pool were not 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 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 show information about this backup set, you can include a wildcard with the name, such as *mybackupset.\** or 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 that 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 nodes 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 might need to have the server convert the file space name that you enter. For example, you might need to have the server convert the name that 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 that are 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 that is 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).

**Important:** Use care when specifying this parameter if multiple node names are also specified. Different nodes might 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
<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	TODAY-7 or -7.  To include files that were active a week ago, specify PITDATE=TODAY-7 or PITDATE=-7



Value	Description	Example
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus <i>days</i> specified.	EOLM-1 To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus <i>days</i> specified.	BOTM+9 To include files that were active on the 10th day of the current month.

### 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
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 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 has 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 that are 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 that are 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.

#### **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:

- 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 that is 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 167. Commands related to GENERATE BACKUPSET

Command	Description
CANCEL PROCESS	Cancels a background server process.
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.

Table 167. Commands related to **GENERATE BACKUPSET** (continued)

<b>Command</b>	<b>Description</b>
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)

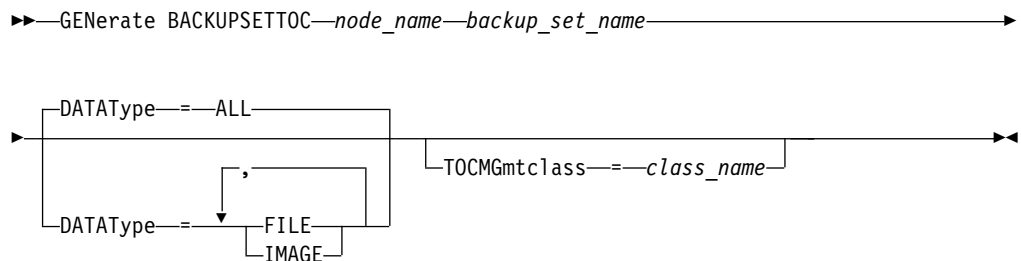
Use this command to generate a table of contents for a backup set that does not already have one. The backup-archive client uses the table of contents to display the backup set, which allows users to select individual files to be restored from the backup set.

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.

#### 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 168. Commands related to GENERATE BACKUPSETTOC*

<b>Command</b>	<b>Description</b>
COPY ACTIVE DATA	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.
UPDATE NODEGROUP	Updates the description of a node group.

---

## GRANT commands

Use the **GRANT** command to grant appropriate privileges or access.

- “GRANT AUTHORITY (Add administrator authority)” on page 554
- “GRANT PROXYNODE (Grant proxy authority to a client node)” on page 558

## GRANT AUTHORITY (Add administrator authority)

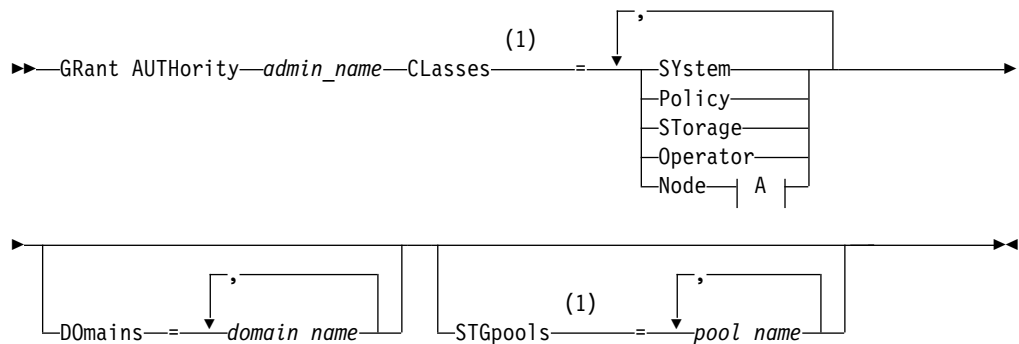
Use this command to grant an administrator one or more administrative privilege classes, and authority to access client nodes.

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.

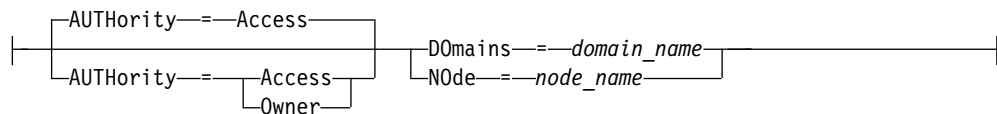
### Privilege class

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.

**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.

#### **NOde**

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**

When used with `CLASSES=POLICY`, 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 169. Commands related to GRANT AUTHORITY

Command	Description
QUERY ADMIN	Displays information about one or more 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 170. 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 1475 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 171. Commands related to **HALT**

Command	Description
CANCEL PROCESS	Cancels a background server process.

Table 171. Commands related to HALT (continued)

<b>Command</b>	<b>Description</b>
CANCEL SESSION	Cancels active sessions with the server.
DISABLE SESSIONS	Prevents new sessions from accessing 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 Tivoli Storage Manager.



*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. The topic number for the command **DEFINE DEVCLASS** for a 3592 device class is 3.13.10.2.

```
help 3.13.10.2
```

### Example: Display help for one command

Display help information about 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 about 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
```



---

## 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 data deduplication, you can specify 0 - 50 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.

This command affects only server-side deduplication processing. In client-side data deduplication processing, duplicates are identified on the backup-archive client.

With the **IDENTIFY DUPLICATES** command, you can start more 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 by 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.

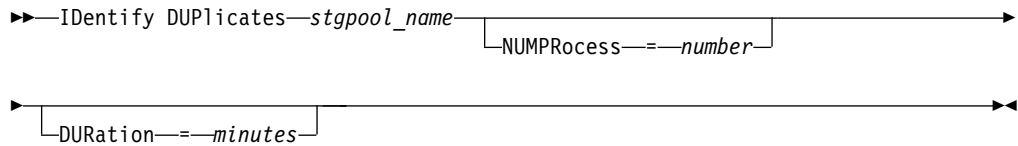
- Duplicate-identification processes can be either active or idle. Processes that are deduplicating files are active. Processes that are waiting for files to deduplicate are idle. Processes remain idle until volumes with data to be deduplicated become available. Processes stop only when canceled or when you change the number of duplicate-identification processes for the storage pool to a value less than what is 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, becomes idle, and then processes five more files, then the total number of files that are processed is nine.

### Privilege class

To issue this command, you must have system privilege.

## 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 - 50 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 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 that 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 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 run indefinitely. They 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 more duplicate-identification processes that run for 60 minutes. At the end of that time, two processes finish the files that 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 server stops idle processes first. If after stopping all idle processes, more processes need to be stopped, the 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 172. Controlling duplicate-identification processes manually

The storage pool definition specifies three duplicate-identification processes. Using the <b>IDENTIFY DUPLICATES</b> command, you specify...	...and a duration of...	The result is...
2 duplicate-identification processes	None specified	One duplicate-identification process finishes the file that it is working on, if any, and then stops. Two processes run indefinitely, or until you reissue the <b>IDENTIFY DUPLICATES</b> command, update the storage pool definition, or cancel a process.
	60 minutes	One duplicate-identification process finishes the file that 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 <b>IDENTIFY DUPLICATES</b> 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 that 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.
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 <b>IDENTIFY DUPLICATES</b> 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.

Table 172. Controlling duplicate-identification processes manually (continued)

<p>The storage pool definition specifies three duplicate-identification processes. Using the <b>IDENTIFY DUPLICATES</b> command, you specify...</p>	<p>...and a duration of...</p>	<p>The result is...</p>
<p>None specified</p>	<p>Not available</p>	<p>The number of duplicate-identification processes resets to the number of processes that are specified in the storage pool definition. This change lasts indefinitely, or until you reissue the <b>IDENTIFY DUPLICATES</b> command, update the storage pool definition, or cancel a process.</p>

### 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 173. 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 CONTENT	Displays information about files in a storage pool volume.
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.

**Important:** For commands that import administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **IMPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

- “IMPORT ADMIN (Import administrator information)” on page 568
- “IMPORT NODE (Import client node information)” on page 572
- “IMPORT POLICY (Import policy information)” on page 579
- “IMPORT SERVER (Import server information)” on page 582

## 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.

**Important:** For commands that import administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **IMPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

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.

**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

This command generates a background process that can be canceled with the **CANCEL PROCESS** command. If an **IMPORT ADMIN** background process is canceled, some of the data is already imported. To display information about background processes, use the **QUERY PROCESS** command.

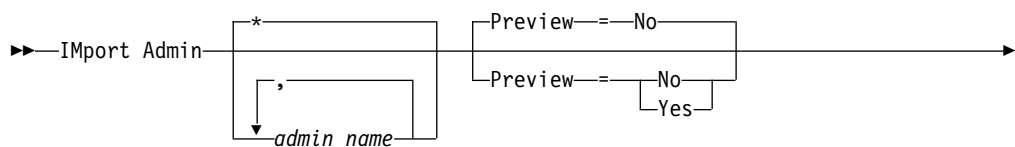
### Restriction:

- If target and source server levels are not compatible, the operation might not work.
- If the administrator definition that is 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 that are being imported can be stored on a CENTERA storage device.

### Privilege class

To issue this command, you must have system privilege.

### Syntax





For this device	Specify
FILE	Any fully qualified file name string. For example:  /imdata/mt1.
SERVER	1 - 250 alphanumeric characters.

### Replacedefs

Specifies whether to replace administrator definitions on the target server. The following parameter values are supported:

**No** Specifies that definitions are not to be replaced.

**Yes**

Specifies that definitions are to be replaced.

The default value is NO.

### 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 that is 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 that is assigned to the MENU1 device class. Issue the command:

```
import admin devclass=menu1 volumenames=file:tapevol
```

### Related commands

Table 174. Commands related to **IMPORT ADMIN**

Command	Description
CANCEL PROCESS	Cancels a background server process.
EXPORT ADMIN	Copies administrative information to external media or directly to another server.
IMPORT NODE	Restores client node information from external media.
IMPORT POLICY	Restores policy information from external media.



Table 174. Commands related to **IMPORT ADMIN** (continued)

<b>Command</b>	<b>Description</b>
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.

**Important:** For commands that import administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **IMPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

If you specify a domain on the source server and if that policy domain also exists on the target server, the imported nodes get associated with that same policy domain 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.

### Restrictions:

1. If target and source server levels are not compatible, the operation might not work.
2. Importing data from a **CENTERA** device class is not supported. However, files that are being imported can be stored on a **CENTERA** storage device.
3. If you use an LDAP directory server to authenticate passwords, any target servers must be configured for LDAP passwords. Data that is imported from a node that authenticates with an LDAP directory server is inaccessible if the target server is not properly configured. If your target server is not configured, imported data from an LDAP node can still go there. But the target server must be configured to use LDAP in order for you to access the imported data.
4. If target and source server levels are not compatible, the operation might not work.
5. You cannot use a **CENTERA** device class as the target medium for an export command, or as the source medium for an import command.
6. Incrementally exporting/importing the following types of client data to another Tivoli Storage Manager server is not supported:
  - VMWare backups where full plus incremental backups need to be periodically, incrementally transferred to another server.
  - Backups groups where full plus differential backups need to be periodically, incrementally transferred to another server.

Full export/import of this data to a new file system on the target is supported by exporting the entire filespace that contains the data. In other words, the export must not use the *FILEDATA=ALLACTIVE*, *FROMDATE*, *TODATE*, or *MERGEFILESACES* options.

The best practice for incrementally transferring this type of data between two servers is to use Node Replication.

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 canceled with the **CANCEL PROCESS** command. If an **IMPORT NODE** background process is canceled, some

of the data might already be imported. To display information about background processes, use the **QUERY PROCESS** command.

For a server that has clients with support for Unicode, you can get the server to convert the file space name that you enter, or use the following parameters:

- **HEXFILESPEC**
- **UNIFILESPEC**

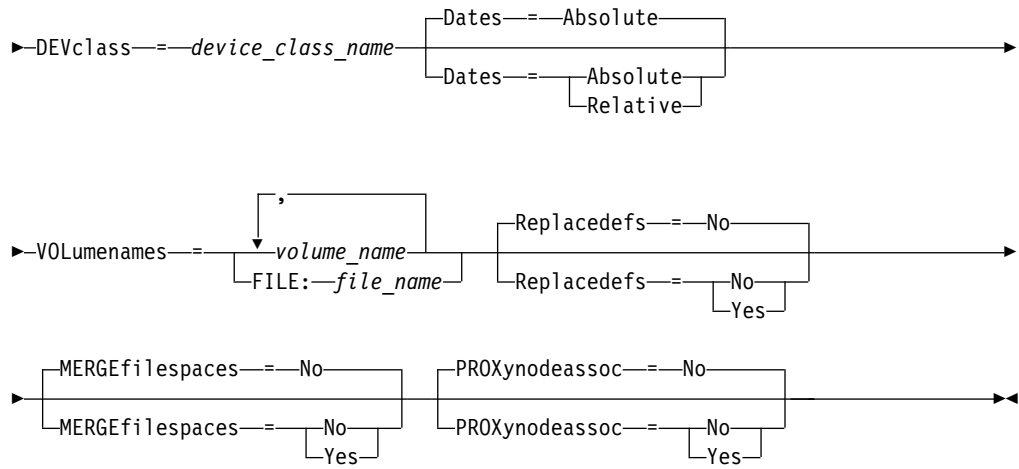
**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

## Privilege class

To issue this command, you must have system privilege.

## Syntax





## 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:**

- Existing file spaces are not replaced. New file spaces are created when identical names are encountered. However, this new name might match an existing name on the client node, which can have file spaces that are not yet backed up to the server.
- This parameter is only 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.

### **D0mains**

Specifies the policy domains from which to import node information. These domains must be 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 can be imported for all nodes that are 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 that is used by the file data is determined by the device class for the storage pool. If it is the same

device class that is 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 parameter supports the following values:

**ALl**

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. The file spaces that are included are both Unicode and non-Unicode.

**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 importing information. The **PREVIEW=YES** option requires that you mount the export volumes. The following values are supported:

**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.

This parameter is optional. The default value is **NO**.

**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 identify duplicates, to make a drive available.

**Dates**

Specifies whether the dates for the file copies are set as the same date when the files were exported, or is adjusted to the import date.

This parameter supports the following values:

**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 import date.

The default value is ABSOLUTE.

If the export media is idle for some time after export, for example; if it is sitting on a shelf for six months, the original backup, or archive dates might be old enough to trigger the file copies to expire immediately when the data is imported into a server. The RELATIVE specification for this value adjusts for time that is 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 five days before the export operation. If the media is saved for six months and then imported, the archive file look like it is inserted six months and five days ago by default, the (DATES=ABSOLUTE) and might expire immediately depending on 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 five days ago during import. The DATES=RELATIVE parameter thus adjusts file backup and archive dates for the time that elapsed since the export operation occurred.

**VOLumentnames (Required)**

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. The parameter supports the following values:

*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 that are 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 you specify volumes that are 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.
SERVER	1 - 250 alphanumeric characters.

**Replacedefs**

Specifies whether to replace definitions on the target server. The default value is NO. The parameter supports the following values:

**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 that the file spaces that are known to the server are Unicode enabled. The server converts the names that 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. A wildcard character can be used to specify a name. This parameter is optional.

#### **MERGEfilespaces**

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 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 exists.

#### **PROXynodeassoc**

Specifies whether 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 that is 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 that are listed in a file named TAPEVOL.

This file contains these lines:

```
TAPE01  
TAPE02  
TAPE03
```

Specify that these tape volumes be read by a device that is 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 175. 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 or directly to another server.
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 import operation might not work.
2. Importing data from a CENTERA device class is not supported. However, files that are 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.

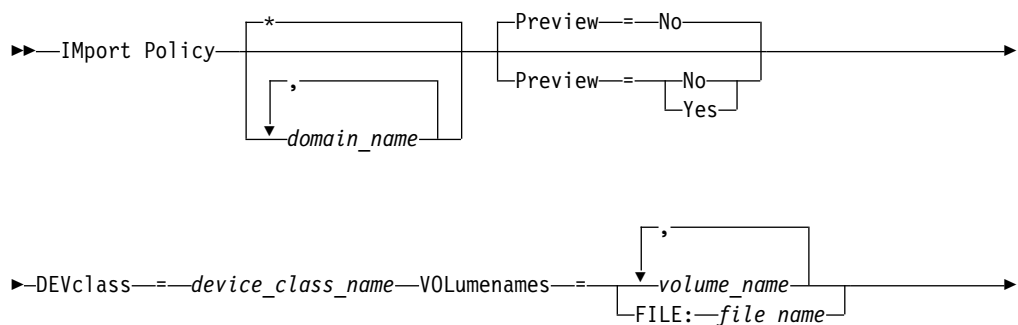
This command generates a background process that can be canceled with the **CANCEL PROCESS** command. If an **IMPORT POLICY** background process is canceled, some of the data is already imported. To display information about background processes, use the **QUERY PROCESS** command.

**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

### 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 without importing information. This parameter supports the following values:

**No** Specifies that the information is to be imported.

### **Yes**

Specifies that the operation is previewed but not completed. Information is reported to the server console and the activity log.

The PREVIEW=YES option requires that you mount the export volumes. This parameter is optional. The default value is NO.

### **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.

### **VOLumentnames (Required)**

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. This parameter supports the following values:

#### *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 you specify volumes that are associated with the following device types:

For this device	Specify
Tape	1 - 6 alphanumeric characters.
FILE	Any fully qualified file name string. For example: /imdata/mt1
SERVER	1 - 250 alphanumeric characters.

## Replacedefs

Specifies whether to replace policy definitions on the target server. This parameter supports the following values:

### Yes

Specifies that objects are to be replaced by the imported objects.

**No** Specifies that objects are not to be replaced by imported objects.

The default value is NO.

## 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 that is 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 that are listed in a file that is named thus:

```
TAPEVOL
TAPEVOL.DATA
```

Specify that these tape volumes be read by a device that is 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 176. Commands related to **IMPORT POLICY**

Command	Description
CANCEL PROCESS	Cancels a background server process.
EXPORT POLICY	Copies policy information to external media or directly to another server.
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 specified client file data from export media to the Tivoli Storage Manager server.

**Important:** For commands that import administrators or nodes, you must consider the method of authentication. The Tivoli Storage Manager server cannot export or import passwords for nodes or administrators that are authenticating with LDAP directory servers. If the current authentication method uses an LDAP directory server and the password is not already synchronized by that server, you must update the password. After issuing the **IMPORT** command, set the password by issuing the **UPDATE ADMIN** or **UPDATE NODE** command.

IBM Tivoli Storage Manager servers with retention protection enabled do not allow import operations.

### Restrictions:

- If target and source server levels are not compatible, the operation might not work.
- Importing data from a CENTERA device class is not supported. However, files that are imported can be stored on a CENTERA storage device.
- If you use an LDAP directory server to authenticate passwords, any target servers must be configured for LDAP passwords. Server data that is exported from a node that authenticates with an LDAP directory server is inaccessible if the target server is not properly configured. If your target server is not configured, exported data from an LDAP node can still go there. But the target server must be configured to use LDAP in order for you to access the data.
- Incrementally exporting or importing the following types of client data to another Tivoli Storage Manager server is not supported:
  - VMware backups where full plus incremental backups need to be periodically, incrementally transferred to another server
  - Backups groups where full plus differential backups must be periodically, incrementally transferred to another server
  - Windows System State data that is periodically, incrementally transferred to another server

Full export or import of this data to a new file system on the target is supported by exporting the entire file space that contains the data. The export must not use the **FILEDATA=ALLACTIVE**, **FROMDATE**, **TODATE**, or **MERGEFILESPPACES** parameters.

Using node replication to incrementally transfer this type of client data between two servers is optimal.

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 canceled with the **CANCEL PROCESS** command. If an **IMPORT SERVER** background process is canceled, some of the data is already imported. To display information about background processes, use the **QUERY PROCESS** command.

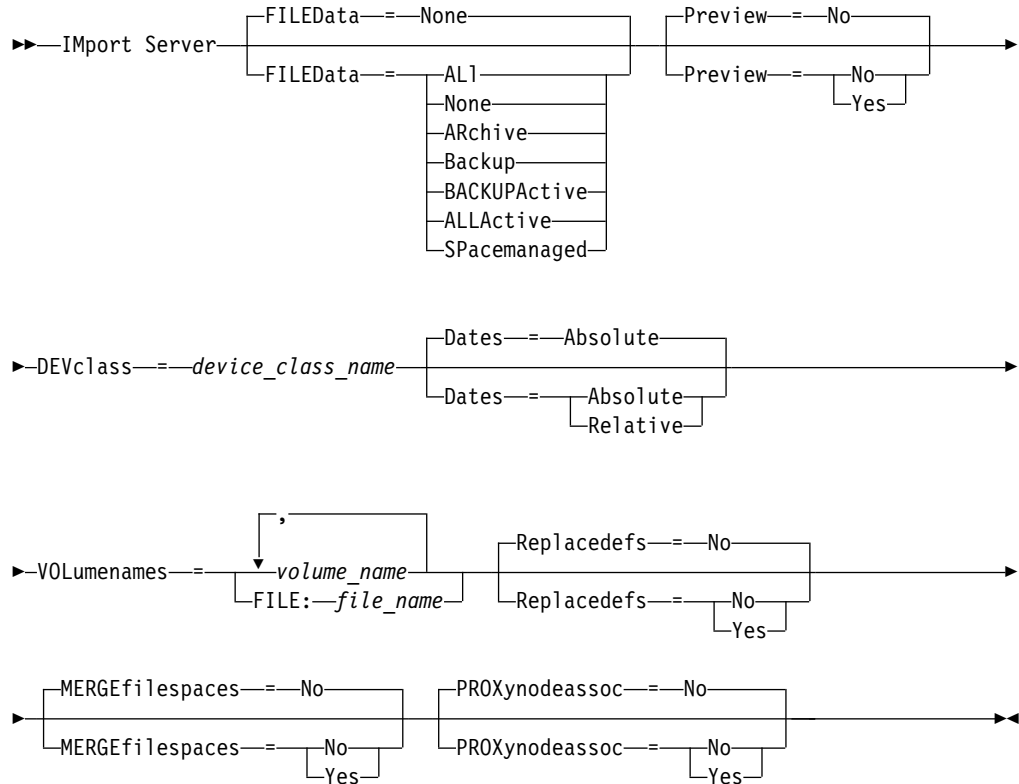
**Limitation:** The Tivoli Storage Manager server does not convert code pages during export, import, and node replication operations. If servers are running in different locales, some information in databases or system output might become unreadable. Invalid characters might be displayed, for example, in the contact information for the administrator and client nodes, and in descriptions of policy domains. Any

field that is stored in the server character set and that includes extended ASCII characters can be affected. To resolve the issue after the import or node replication operation, update the fields with the appropriate **UPDATE** commands. This server limitation does not affect client data. Any client data that was exported, imported, or replicated can be restored, retrieved, and recalled.

## Privilege class

To issue this command, you must have system privilege.

## Syntax



## Parameters

### FILEData

Specifies the type of files that can be imported for all nodes that are defined to the server. This parameter is optional. The default value is NONE.

The device class that is used to access the file data is determined by the device class for the storage pool. If it is the same device class that is 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. This parameter supports the following values:

**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. This parameter supports the following values:

**No** Specifies that the server information is to be imported.

**Yes**

Specifies that the operation is previewed but not completed. Information is transferred to the server console and the activity log.

This parameter is optional. The default value is NO. If the PREVIEW=YES option is specified, you must mount the export volumes.

**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 for the file copies are set as the same date when the files were exported, or is adjusted to the import date.

If the import media is idle for some time after export, for example; if it is sitting on a shelf for six months, the original backup, or archive dates might be old enough to trigger the file copies to expire immediately when the data is imported into a server. The RELATIVE specification for this value adjusts for time that is 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 five days before the export operation. If the export media are saved for six months and then imported, the archive file looks like it is inserted six months and five days ago by default (DATES=ABSOLUTE) and might expire immediately 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 five days ago during import. DATES=RELATIVE parameter thus adjusts file backup and archive dates for the time that elapsed since the export operation occurred.

This parameter supports the following values:

**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.

The default value is ABSOLUTE.

**VOLumentnames (Required)**

Specifies the volumes to be used for the import operation. Volumes must be imported in the same order as they were exported. This parameter supports the following values:

*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 that are 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 you specify volumes that are associated with the following device types:

For this device	Specify
Tape	1 - 6 alphanumeric characters.
FILE	Any fully qualified volume or file name string. An example is /imdata/mt1.
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. This parameter supports the following values:

**No** Specifies that objects are not to be replaced by imported objects.

**Yes**

Specifies that objects are to be replaced by the imported objects.

The default value is NO.

**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. This parameter supports the following values:

**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 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 exists on the target server.

The default is NO.

**PROXynodeassoc**

Specifies whether 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 that is 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 that is 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 exist.

```
import server devclass=menu1 volumenames=tape01,tape02,tape03 mergefilespaces=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 that are listed in a file named TAPEVOL. Specify that the tape volumes are read by a device that is 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 177. Commands related to IMPORT SERVER*

Command	Description
CANCEL PROCESS	Cancels a background server process.
COPY ACTIVE DATA	Copies active backup data.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
IMPORT ADMIN	Restores administrative information from external media.



Table 177. Commands related to **IMPORT SERVER** (continued)

<b>Command</b>	<b>Description</b>
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.

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

```
▶▶—INsert MACHine—machine_name—sequence_number—————▶▶
|
|—CHARacteristics—==—text—————▶▶
|—RECOVERyInStructions—==—text————▶▶
```

### 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 178. Commands related to **INSERT MACHINE**

Command	Description
DEFINE MACHINE	Defines a machine for DRM.
DELETE MACHINE	Deletes a machine.
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 `backscript` 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 `backscript` on the client machine and issues messages based on the return code from `backscript`.

```
issue message i "Starting backup"
define clientaction nodename action=command objects="c:\backscript" 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 179. Commands related to *ISSUE MESSAGE*

Command	Description
COPY SCRIPT	Creates a copy of a script.
DEFINE SCRIPT	Defines a script to the 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 pre-labeled.

**Restriction:** Use this command only for MANUAL, SCSI, ACSLS, and 349X libraries. The 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 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.

For detailed and current drive and library support information, see the Supported Devices website for your operating system:

[http://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)

To use the **LABEL LIBVOLUME** command, at least one drive must exist that is not in use by another Tivoli Storage Manager process. This includes idle volumes that are mounted. If necessary, use the **DISMOUNT VOLUME** command to dismount the idle volume to make that drive available.

By default, the **LABEL LIBVOLUME** command does not overwrite an existing label. However, if you want to overwrite an existing label, you can specify the **OVERWRITE=YES** option.

**Attention:**

- By overwriting a volume label, you destroy all data on the volume. Use caution when you overwrite volume labels to avoid deleting valid data.
- The labels on VolSafe volumes can be overwritten only once. Therefore, use the **LABEL LIBVOLUME** command only once for VolSafe volumes. You can guard against overwriting the label by using the **OVERWRITE=NO** option with the **LABEL LIBVOLUME** command.

When you use the **LABEL LIBVOLUME** command, you can identify the volumes to be labeled in one of the following ways:

- Explicitly name one volume.
- Enter a range of volumes by using the **VOLRANGE** parameter.
- Use the **VOLLIST** parameter to specify a file that contains a list of volume names or to explicitly name one or more volumes.

For automated libraries, you are prompted to insert the volume in the entry/exit slot of the library. If no I/O convenience station is available, insert the volume into an empty slot. For manual libraries, you are prompted to load the volume directly into a drive.

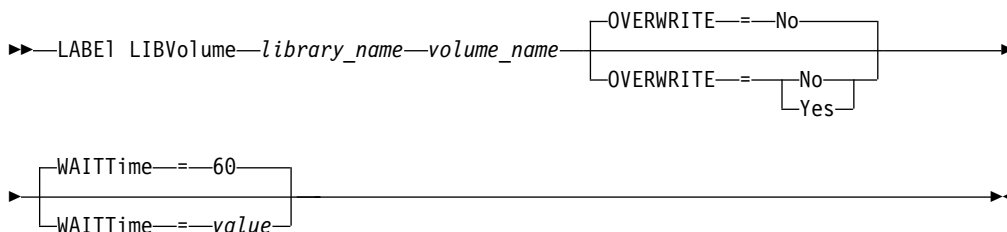
**Tip:** To automatically label tape volumes, you can use the **AUTOLABEL** parameter on the **DEFINE LIBRARY** and **UPDATE LIBRARY** commands. By using the **AUTOLABEL** parameter, you eliminate the need to pre-label a set of tapes. This method is more efficient than using the **LABEL LIBVOLUME** command, which requires you to mount volumes separately. If you use the **AUTOLABEL** parameter with a SCSI library, you

must check in tapes by specifying CHECKLABEL=BARCODE on the **CHECKIN LIBVOLUME** command. The **AUTOLABEL** parameter defaults to YES for all non-SCSI libraries and to NO for SCSI libraries.

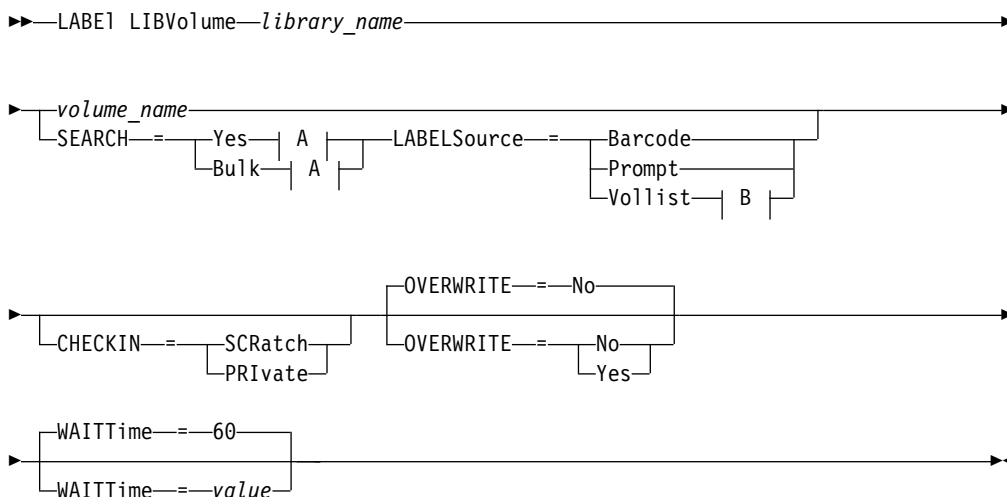
## Privilege class

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

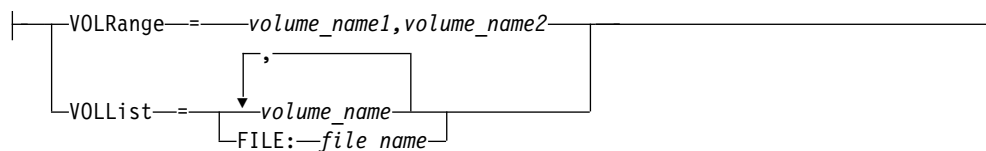
## Syntax for a manual library



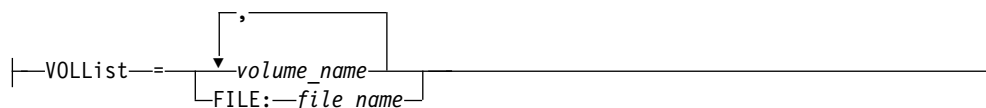
## Syntax for a SCSI library



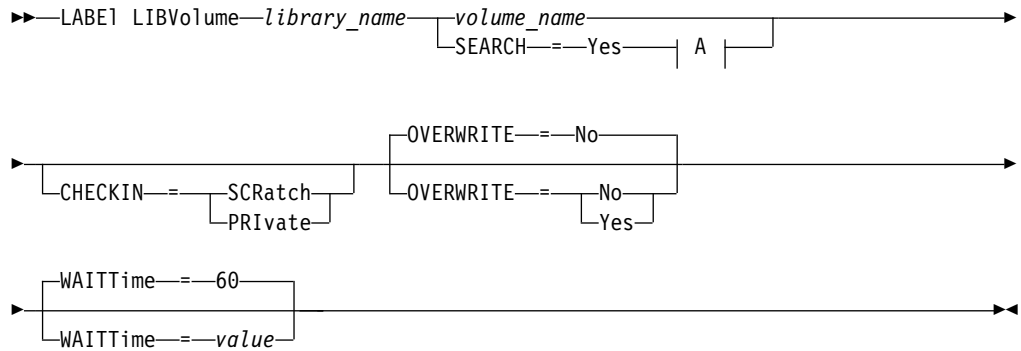
### A (SEARCH=Yes, SEARCH=Bulk):



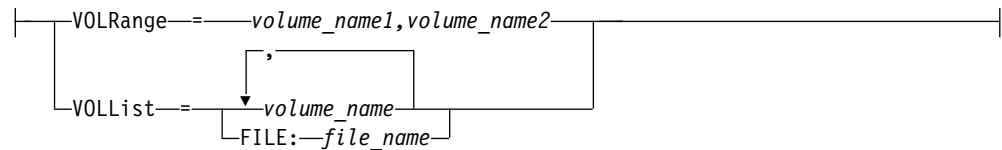
### B (LABELSource=Vollist):



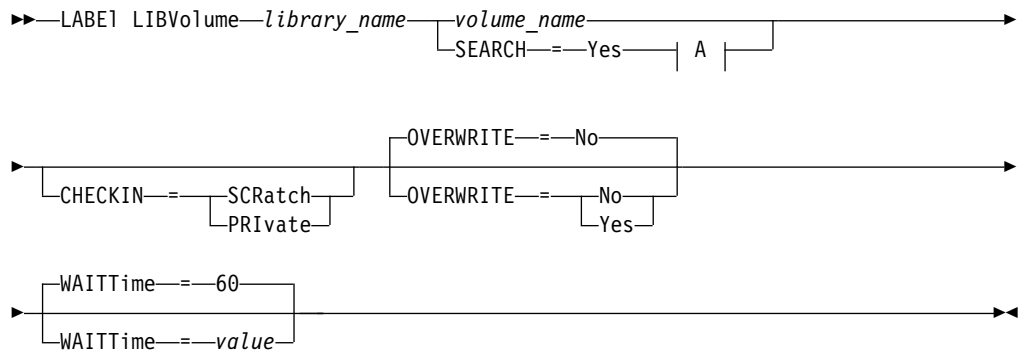
## Syntax for a 349X library



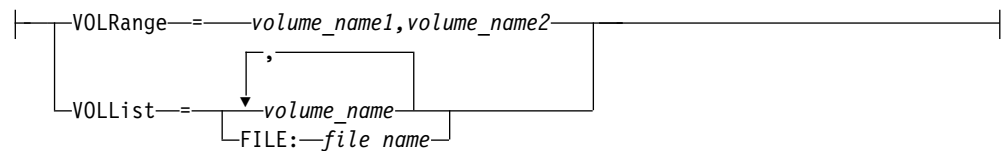
### A (SEARCH=Yes):



## 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 is 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 is labeled.

**Note:** If you specify a volume name, the name you specify overrides the label that is printed on the cartridge.

- For MANUAL libraries: The server requests that the volume is 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 is not labeled, and a message is 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, the command labels the volume, but does 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.

The following values are valid:

#### **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.

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 that contains 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. Then, 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 that are 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 that are supported by Tivoli Storage Manager and that 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 that are 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:

<b>Parameter</b>	<b>Description</b>
<code>volrange=bar110,bar130</code>	The 21 volumes are labeled: bar110, bar111, bar112,...bar129, bar130.
<code>volrange=bar11a,bar13a</code>	The 3 volumes are labeled: bar11a, bar12a, bar13a.
<code>volrange=123400,123410</code>	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.

The following values are valid:

*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 that is 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.

You can specify the following values:

**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 does not label the volume and displays a message.

**Important:** For bar code support to work properly, the appropriate device drivers must be installed for the libraries.

**VolList**

This option applies only to SCSI libraries. The server attempts to read the specified file or list of files. If the attempt fails, the server does not label the volumes and displays a message.

**OVERWRITE**

Specifies whether the server attempts to overwrite existing labels. This parameter is optional. The default is NO. You can specify the following values:

**No** Specifies that the server labels only unlabeled volumes. For StorageTek VolSafe 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 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 that 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 wait 60 minutes for you to reply. Alternatively, suppose that you specify a wait time of 0. If you inserted a tape, a wait time of zero causes the operation to continue without prompting. If you did not insert a tape, a wait time of zero causes the operation to fail.

**Example: Automatically label library volumes**

Label tapes in a SCSI library named AUTO automatically as you are checking in the volumes.

```
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 three volumes are labeled: bar11a, bar12a, bar13a.

```
label libvolume abc checkin=scratch search=yes volrange=bar11a,bar13a  
labelsource=barcode
```

## Related commands

Table 180. 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.



## Related commands

Table 181. Commands related to **LOAD DEFALERTTRIGGERS**

Command	Description
"DEFINE ALERTTRIGGER (Define an alert trigger)" on page 124	Associates specified messages to an alert trigger.
"DELETE ALERTTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.

---

## LOCK commands

Use the **LOCK** command to prevent users from accessing the server.

- “LOCK ADMIN (Lock out an administrator)” on page 601
- “LOCK NODE (Lock out a client node)” on page 603
- “LOCK PROFILE (Lock a profile)” on page 605

## LOCK ADMIN (Lock out an administrator)

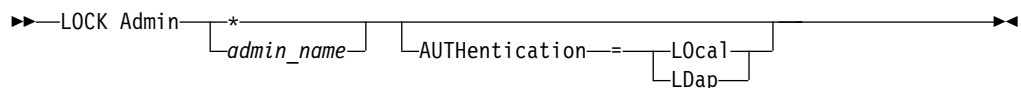
Use this command to prevent an administrator from accessing the server. The administrator is locked out until a system administrator uses the **UNLOCK ADMIN** command to reestablish access for the administrator.

You can use the authentication filter to lock all administrators, excluding console administrators. After configuring an LDAP directory server for password authentication, you can lock administrators to force them to create passwords that authenticate with an LDAP server.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *admin\_name* (Required)

Specifies the name of the administrator to be locked out. You can use wildcard characters to specify the administrator name. You do not have to enter an administrator name if you want to lock all of the administrators according to their authentication method. Use the wildcard with an authentication method to lock multiple administrators.

#### **AUTHentication**

Specifies the method of authentication that the administrator uses to log in.

#### **Local**

Specifies to lock administrators who authenticate to the IBM Tivoli Storage Manager server.

#### **Ldap**

Specifies to lock administrators who authenticate to the LDAP directory server.

### Example: Lock out an administrator

Lock out the administrator CLAUDIA. Issue the command:

```
lock admin claudia
```

### Example: Lock out all administrators who authenticate to the Tivoli Storage Manager server database

Use the wildcard character (\*) to lock all the administrators who authenticate their passwords locally. Console administrators are not affected by this command. Issue the following command:

```
lock admin * authentication=local
```

## Related commands

*Table 182. Commands related to LOCK ADMIN*

<b>Command</b>	<b>Description</b>
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.
UNLOCK ADMIN	Enables a locked administrator to access Tivoli Storage Manager.



## LOCK NODE (Lock out a client node)

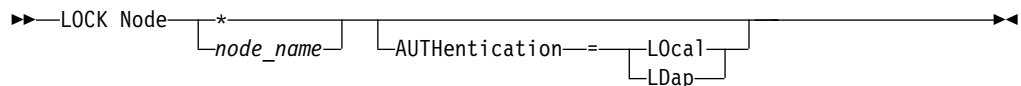
Use this command to prevent a client node from accessing the server. A locked client node cannot perform any Tivoli Storage Manager operations, even if the operations are scheduled.

After configuring an LDAP directory server for password authentication, you can lock nodes to force them to use passwords that authenticate with an LDAP server.

### Privilege class

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

### Syntax



### Parameters

#### *node\_name*

Specifies the name of the client node to lock out. You can use a wildcard character instead of a node name if you want to lock all of the nodes according to their method of authentication.

#### **AUTHentication**

Specifies the method of password authentication that is needed to log into a node.

#### **L0cal**

Specifies to lock nodes that authenticate with the Tivoli Storage Manager server.

#### **LDap**

Specifies to lock nodes that authenticate with an LDAP directory server.

### Example: Lock a specific client node

Lock the client node SMITH.

```
lock node smith
```

### Example: Lock all nodes that authenticate to the local Tivoli Storage Manager database

Issue the following command to lock all nodes that authenticate with the Tivoli Storage Manager server:

```
lock node * authentication=local
```

### Related commands

Table 183. Commands related to LOCK NODE

Command	Description
QUERY NODE	Displays partial or complete information about one or more clients.

*Table 183. Commands related to LOCK NODE (continued)*

<b>Command</b>	<b>Description</b>
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 [60]
                               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 184. 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 184. Commands related to LOCK PROFILE (continued)*

<b>Command</b>	<b>Description</b>
UNLOCK PROFILE	Enables a locked profile to be distributed to managed servers.
UPDATE PROFILE	Changes the description of a profile.



## 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 185. 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 that are 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.

If you use this command to start a migration process, but the storage pool does not have a next storage pool identified in the hierarchy, a reclamation process is triggered for the source storage pool. To prevent the reclamation process, define the next storage pool in the hierarchy. Then, start the migration process.

The **MIGRATE STGPOOL** command honors the values of the following parameters on the **DEFINE STGPOOL** and **UPDATE STGPOOL** commands:

- MIGPROCESS
- MIGDELAY
- MIGCONTINUE
- NEXTPOOL
- LOWMIG

**Tip:** You can override the value of the **LOWMIG** parameter on **DEFINE STGPOOL** and **UPDATE STGPOOL** by specifying a value for the **LOWMIG** parameter on the **MIGRATE STGPOOL** command.

The **MIGRATE STGPOOL** command ignores the value of the **HIGHMIG** parameter of the storage pool definition. Migration occurs regardless of the value of the **HIGHMIG** parameter.

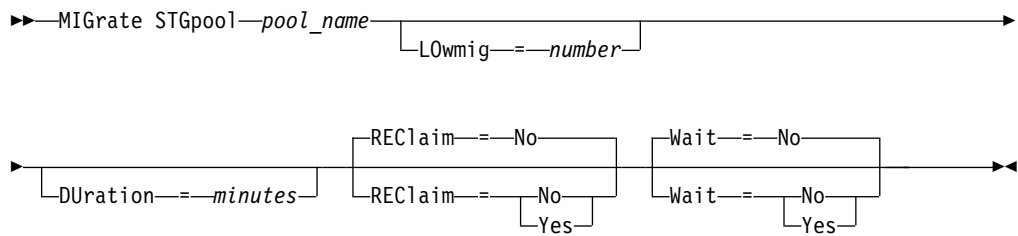
This command creates one or more migration processes that can be canceled with the **CANCEL PROCESS** command. The number of processes is limited by the **MIGPROCESS** attribute of the storage pool definition. To display information about background processes, use the **QUERY PROCESS** command.

**Remember:** Migrating data from a primary storage pool that is set up for data deduplication to another primary storage pool that is also set up for data 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



## 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 runs before being automatically canceled. 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 end. As a result, the migration might 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 migration should stop when the amount of data in the pool is at or below this percentage of the pool's estimated capacity. This parameter is optional.

The calculation for sequential-access disk storage pools includes the capacity of all the scratch volumes that are specified for the pool. Because migration is by node or filesystem, depending upon collocation, the occupancy of the storage pool can fall below the value that you specified for this parameter. To empty the storage pool, set LOWMIG=0. For other types of sequential-access storage pools, 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. You can specify a number from 0 to 99 for this optional parameter. The default value is the LOWMIG attribute of the storage pool definition.

### **RECLAIM**

Specifies whether reclamation is attempted for the storage pool before completing 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 before starting the migration.

#### **Yes**

Specifies that the server will attempt reclamation before 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 before completing the migration. If no volumes meet the reclamation threshold or if, after reclamation, the LOWMIG threshold has not been reached, the server will begin the migration. Before reclaiming space for storage pools defined with



RECLAMATIONTYPE=SNAPLOCK, the server deletes 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 that are 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 before 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 186. 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.

- “MOVE DATA (Move files on a storage pool volume)” on page 613
- “MOVE DRMEDIA (Move disaster recovery media offsite and back onsite)” on page 617
- “MOVE GRPMEMBER (Move a server group member)” on page 633
- “MOVE MEDIA (Move sequential-access storage pool media)” on page 634
- “MOVE NODEDATA (Move data by node in a sequential access storage pool)” on page 642

## 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, you can use this command to 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.

You cannot move data into or out of a storage pool that is 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 might 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 or another copy storage pool. These files are then written to the destination volumes in the original copy storage pool or active-data pool.

During the data movement process, active-data pools cannot be used to obtain data.

If you run the **MOVE DATA** command on an offsite volume that contains collocated data, it might be necessary to issue the **MOVE DATA** command multiple times to move all of the data out of the volume. For example, if you are using filespace collocation groups with an offsite volume that contains filespace in a collocation group and filespace that are not in the group, you must issue two **MOVE DATA** commands. Each **MOVE DATA** command moves the data for a single collocated or non-collocated group of files.

Do not use the **MOVE DATA** command if a restore process (**RESTORE STGPOOL** or **RESTORE VOLUME**) is running. The **MOVE DATA** command might 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 must reissue the **MOVE DATA** command after the restore operation completes in order to move any remaining files.

**Remember:**

Issuing this command removes duplicate data when:

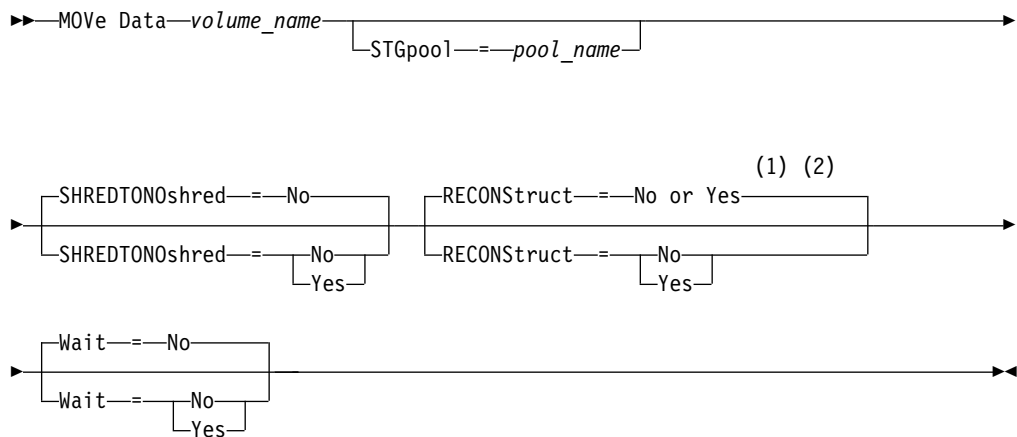
- Moving data from a primary storage pool that is set up for data deduplication to another primary storage pool that is also set up for data deduplication.
- Moving data within a copy storage pool that is set up for data deduplication.
- Moving data within an active-data pool that is set up for data deduplication.

A volume in a deduplicated storage pool might contain files that are logically deleted but are still linked by files on other volumes. If you use the **MOVE DATA** command to move the contents of a deduplicated storage pool volume to a non-deduplicated storage pool, the logically deleted files are not written to the new volume since they do not exist logically. The deleted files are kept on the original volumes for other files to reference. The **MOVE DATA** process ends successfully but none of the deleted files are moved to the new target volume and the source volume is not deleted. You can issue the **QUERY CONTENT** command with the **FOLLOWLINKS=YES** or **FOLLOWLINKS=JUSTLINKS** parameter to verify whether the volume contains files that are linked by files on other volumes.

**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 is 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 fails.

#### Yes

Specifies that the server allows data to be moved from a storage pool that enforces shredding to a storage pool that does not enforce shredding. The source data is 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 any of the following conditions are true:

- The data format is NETAPPDUMP, CELERRADUMP, or NDMPDUMP.
- The data is in a storage pool that is configured for data deduplication.
- The target storage pool for the data movement is configured for data deduplication.

**Attention:** Reconstruction removes inactive backup files in active-data pools. If you specify RECONSTRUCT=NO when moving the data in an active-data pool that is not configured for data deduplication, inactive backup files remain in the storage pool.

Possible values are:

**No** Specifies that reconstruction of file aggregates is not completed during data movement.

#### Yes

Specifies that reconstruction of file aggregates is completed during data movement. You can only specify this option when both the source and the target storage pools are sequential-access.

### 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 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.

### 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 187. 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 onsite and offsite.
QUERY ACTLOG	Displays messages from the server activity log.
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 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 volumes that are to be moved offsite and to identify the expired or empty volumes that are to be moved onsite. You can track database backup volumes, and volumes in copy storage pools and active-data storage pools.

The processing of volumes by this command depends on what the volumes are used for:

### Backups of the server database

To control whether the command processes database backup volumes, use the **SOURCE** parameter on this command. The command can process volumes that are used for full plus incremental or snapshot database backups. You cannot specify virtual volumes (backup objects that are 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.

### Copy storage pools

The **MOVE DRMEDIA** command always processes copy storage-pool volumes.

### Active-data storage pools

By default, volumes in active-data storage pools are not eligible for processing by the **MOVE DRMEDIA** command. To process active-data pool volumes, you must issue the **SET DRMACTIVEDATASTGPOOL** command first, or specify the **ACTIVEDATASTGPOOL** parameter on the **MOVE DRMEDIA** command.

You can use the **QUERY ACTLOG** command to see whether the **MOVE DRMEDIA** command was successful. You can also view this information from the server console.

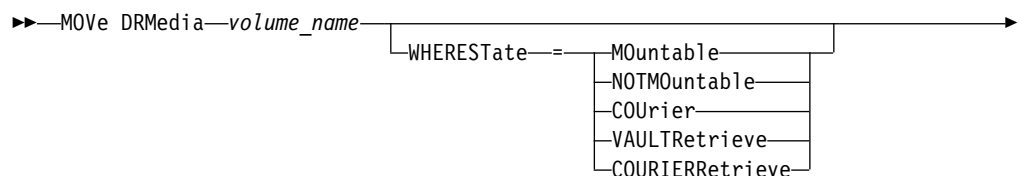
**Restriction:** Do not run the **MOVE DRMEDIA** and **BACKUP STGPOOL** commands concurrently. Ensure that the storage pool backup processes are complete before you issue the **MOVE DRMEDIA** command.

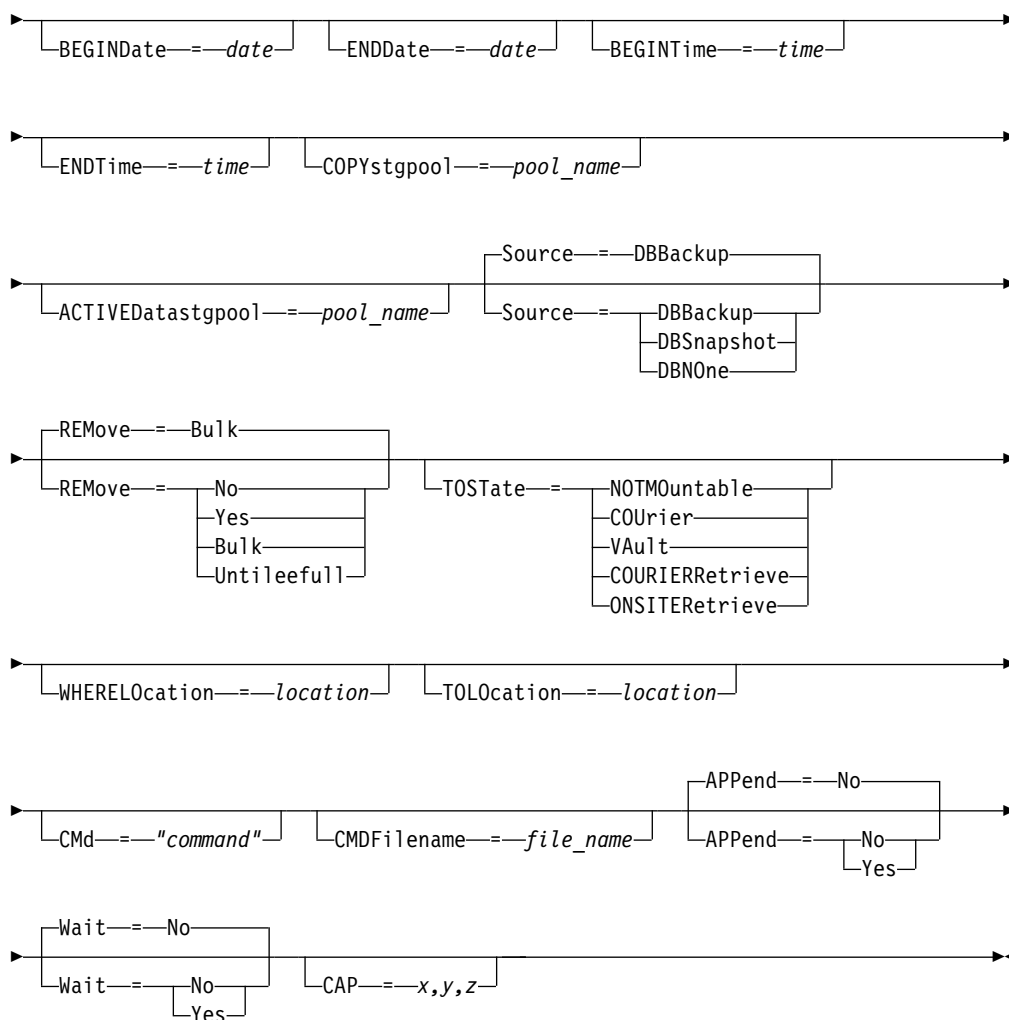
## Privilege class

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 volume to be processed. You can use wildcard characters. If you use wildcard characters to specify this name, you must also specify the **WHERESTATE** parameter. The server looks for matching names among the following eligible volumes:

- Database backup volumes, as specified by the **SOURCE** parameter of this command.
- Copy storage pool volumes from the storage pools named in the **COPYSTGPOOL** parameter. If you do not use the **COPYSTGPOOL** parameter, the server processes volumes from copy storage pools that were previously specified in the **SET DRMCOPYSTGPOOL** command.
- Active-data storage pool volumes from the storage pools named in the **ACTIVEDATASTGPOOL** parameter. If you do not use the **ACTIVEDATASTGPOOL** parameter, the server processes volumes from active-data storage pools that were previously specified in the **SET DRMACTIVEDATASTGPOOL** command.

Other parameters can also limit the results of the command.

### **WHERESTATE**

Specifies the state of volumes to be processed. This parameter is required if the **TOSTATE** parameter is not specified or if you use a wildcard character in the



volume name. For more information, see Table 189 on page 627 and Table 190 on page 628. Specify one of the following values:

**MOUNTable**

These volumes contain valid data and are available for onsite processing. The values change to NOTMOUNTABLE if the **TOSTATE** parameter is not specified.

Depending on the outcome of the **REMOVE** parameter, the server might eject volumes in an automated library before you change 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 whether the volumes are ejected from the library.

**NOTMOUNTable**

These volumes are onsite, contain valid data, and are not available for onsite processing. The values change to COURIER if the **TOSTATE** parameter is not specified.

**COURier**

These volumes are with the courier and being moved offsite. The values change only to VAULT.

**VAULTRetrieve**

These volumes are at the offsite vault and do not contain valid data. The values change to COURIERRETRIEVE if the **TOSTATE** parameter is not specified.

**COURIERRetrieve**

These volumes are with the courier and being moved onsite. The values change only to ONSITERETRIEVE. The server deletes the volume records of the database backup and scratch copy storage pool volumes from the database.

**BEGINDate**

Specifies the beginning date that is used to select volumes. This parameter is optional. Volumes are considered eligible if the **MOVE DRMEDIA** command changes 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 by using one of the following values:

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.	TODAY-7 or -7  To identify volumes that were changed to their current state a week ago, you can specify TODAY-7 or -7.
<b>EOLM</b> (end of last month)	The last day of the previous month.	EOLM
<b>EOLM-days</b>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.

Value	Description	Example
<b>BOTM</b> (beginning of this month)	The first day of the current month.	BOTM
<b>BOTM+days</b>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### ENDDate

Specifies the ending date that is used to select volumes. This parameter is optional. Volumes are considered eligible if the **MOVE DRMEDIA** command changes the volume to its current state on or before the specified date. The default is the current date.

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

Value	Description	Example
<i>MM/DD/YYYY</i>	A specific date.	09/15/1998
<b>TODAY</b>	The current date.	TODAY  To identify volumes that were changed to their current state today, specify TODAY.
<b>TODAY-days</b> or <i>-days</i>	The current date minus days specified. The maximum number of days is 9999.	TODAY-1 or -1  To identify volumes that were changed to their current state a week ago, you can specify TODAY-1 or -1.
<b>EOLM</b> (end of last month)	The last day of the previous month.	EOLM
<b>EOLM-days</b>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
<b>BOTM</b> (beginning of this month)	The first day of the current month.	BOTM
<b>BOTM+days</b>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### BEGINTime

Specifies the beginning time that is used to select volumes for processing. This parameter is optional. Volumes are considered eligible if the **MOVE DRMEDIA** command changes the volume to its current state on or after the specified time and date. The default is midnight (00:00:00) on the date that is specified with the **BEGINDATE** parameter.

You can specify the time by 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
<b>NOW</b>	The current time on the specified begin date.	NOW
<b>NOW+HH:MM</b> or <b>+HH:MM</b>	The current time plus hours and minutes on the specified begin date.	NOW+03:00 or +03:00
<b>NOW-HH:MM</b> or <b>-HH:MM</b>	The current time minus hours and minutes on the specified begin date.	NOW-03:30 or -03:30  If you issue the <b>MOVE DRMEDIA</b> command at 9:00 with <b>BEGINTIME=NOW-03:30</b> or <b>BEGINTIME=-03:30</b> , the server identifies the volumes that were changed to their current state at 5:30 on the begin date that you specify.

### **ENDTime**

Specifies the ending time that is used to select volumes for processing. This parameter is optional. Volumes are considered eligible if the **MOVE DRMEDIA** command changes 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 by 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
<b>NOW</b>	The current time on the specified end date.	NOW
<b>NOW+HH:MM</b> or <b>+HH:MM</b>	The current time plus hours and minutes on the specified end date.	NOW+03:00 or +03:00  If you issue the <b>MOVE DRMEDIA</b> command at 9:00 with <b>ENDTIME=NOW+03:30</b> or <b>ENDTIME=+03:30</b> , the server identifies the volumes that were changed to their current state at 12:30 on the end date you specify.
<b>NOW-HH:MM</b> or <b>-HH:MM</b>	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 the **WHERESTATE** parameter.

The copy storage pools that are specified with this parameter override copy storage pools that are specified with the **SET DRMCOPYSTGPOOL** command. If this parameter is not specified, the server selects the storage pools as follows:

- If the **SET DRMCOPYSTGPOOL** command was previously issued with valid copy storage pool names, the server processes only those storage pools.
- If the **SET DRMCOPYSTGPOOL** command was not issued, or if all of the copy storage pools are removed by using the **SET DRMCOPYSTGPOOL** command, the

server processes all copy storage pool volumes in the specified state. The states available are MOUNTABLE, NOTMOUNTABLE, COURIER, VAULTRETRIEVE, or COURIERRETRIEVE.

#### **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 active-data pools that are specified with the **SET DRMACTIVEDATASTGPOOL** command. If this parameter is not specified, the server selects the storage pools in the following way:

- If the **SET DRMACTIVEDATASTGPOOL** command was previously issued with valid active-data pool names, the server processes only those storage pools.
- If the **SET DRMACTIVEDATASTGPOOL** command was not issued, or all of the active-data pools are removed by using the **SET DRMACTIVEDATASTGPOOL** command, the server processes all active-data pool volumes in the specified state. The states available are NOTMOUNTABLE, COURIER, VAULTRETRIEVE, or COURIERRETRIEVE. Volumes in the MOUNTABLE state are not processed.

#### **Source**

Specifies whether to include database backup volumes for processing. This parameter is optional. The default is DBBACKUP. Specify one of the following values:

##### **DBBackup**

Specifies that the server includes full and incremental database backup volumes for processing.

##### **DBSnapshot**

Specifies that the server includes database snapshot backup volumes for processing.

##### **DBNone**

Specifies that the server does not include any database backup volumes for processing.

#### **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, NO, BULK, and UNTILEEFULL. The default is BULK. The response of the server to each value and the default value depends on the type of library.

**Restriction:** You can use the **REMOVE=UNTILEEFULL** option only with the library type SCSI.

#### **SCSI libraries**

The response of the server to the command depends on whether the library has entry/exit ports, and if so, whether a port is available for use. See the following table.

Table 188. Server response for SCSI libraries

Library characteristic	Server response when you specify REMOVE=YES	Server response when you specify REMOVE=BULK	Server response when you specify REMOVE=NO	Server response when you specify REMOVE=UNTILEEFULL
Library has no entry/exit ports	The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.  The server then prompts you to remove the cartridge from the slot and to issue a <b>REPLY</b> command.	The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.  The server does not prompt you to remove the cartridge and does not require a <b>REPLY</b> command.	The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.  The server does not prompt you to remove the cartridge and does not require a <b>REPLY</b> command.	The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.  The server does not prompt you to remove the cartridge and does not require a <b>REPLY</b> command.
Library has entry/exit ports and an entry/exit port is available	The server moves the cartridge to the available entry/exit port and specifies the port address in a message.  The server then prompts you to remove the cartridge from the slot and to issue a <b>REPLY</b> command.	The server moves the cartridge to the available entry/exit port and specifies the port address in a message.  The server does not prompt you to remove the cartridge and does not request a <b>REPLY</b> command.	The server specifies the port address in a message.  The server does not prompt you to remove the cartridge and does not request a <b>REPLY</b> command.	The server moves the cartridge to the available entry/exit port and specifies the port address in a message.  The server does not prompt you to remove the cartridge and does not request a <b>REPLY</b> command.
Library has entry/exit ports, but no ports are available	The server leaves the cartridge in its current slot within the library and specifies the slot address in a message.  The server then prompts you to remove the cartridge from the slot and to issue a <b>REPLY</b> command.	The server waits for a port to be made available.	The server specifies the port address in a message.  The server does not prompt you to remove the cartridge and does not request a <b>REPLY</b> command.	The command fails and any remaining eligible volumes are not processed.  Make the port available and issue the command again.

### 349X libraries

#### REMOVE=YES

The 3494 Library Manager ejects the cartridge to the convenience I/O station.

#### REMOVE=BULK

The 3494 Library Manager ejects the cartridge to the high-capacity output facility.

#### REMOVE=NO

The 3494 Library Manager does not eject the volume. The server leaves the cartridge in the library in the INSERT category for use by other applications.

### ACSLs libraries

**REMOVE=YES or REMOVE=BULK**

The server ejects the cartridge to the convenience I/O station.

The server then deletes the volume entry from the server library inventory.

When you move 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.

**REMOVE=NO**

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.

**External libraries**

You can specify **REMOVE=YES**, **REMOVE=BULK**, or **REMOVE=NO**. For any value, the server requests the external library manager to eject the volume from the library.

It depends on the external library manager whether the volume is ejected from the library. Refer to the external library documentation for information about the procedures to follow when you use the **MOVE DRMEDIA** command to track volumes.

**TOSTate**

Specifies the destination state of the volumes that are processed. This parameter is required if the **WHERESTATE** parameter is not specified. If you specify **TOSTATE** parameter but not **WHERESTATE** parameter, you must specify the volume name. Wildcard characters are not allowed. See Table 189 on page 627 and Table 190 on page 628.

Specify one of the following values:

**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, the server might eject the volumes from the library before you change them to the NOTMOUNTABLE state, depending on the behavior of the **REMOVE** parameter.

For external libraries, the server sends requests to the external library manager to eject the volumes. Whether the volumes are ejected from the library depends on the external library manager. Refer to the external library documentation for information about the procedures to follow when you use 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, the server might eject the volumes from the library before you change them to the COURIER state.

For external libraries, the server sends requests to the external library manager to eject the volumes. Whether the volumes are ejected from the library depends on the external library manager. Refer to the external library documentation for information about the procedures to follow when you use 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, the server might eject the volumes from the library before you change them to the VAULT state.

For external libraries, the server sends requests to the external library manager to eject the volumes. Whether the volumes are ejected from the library depends on the external library manager. Refer to the external library documentation for information about the procedures to follow when you use 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, the server 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 that is 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 that is defined by the **SET DRMNOMOUNTABLE** command is used.

### **CMd**

Specifies a command to be issued for each volume that is processed by the **MOVE DRMEDIA** command. DRM writes the commands to a file that is specified by the **CMDFILENAME** parameter. After the MOVE DRMEDIA operation is completed, the commands in the file can be issued. The command can contain up to 255 characters. If the command contains more than 240 characters, it is split into multiple lines, and continuation characters (+) are added. You might need to alter the continuation character based on the operating system. This parameter is optional.

#### *command*

The command string that is enclosed in quotation marks. The string must not include embedded quotation marks. For example, the following **CMd** parameter is valid:

```
cmd="checkin libvol lib8mm &vol status=scratch"
```

The following example is not a valid way to specify the **CMd** parameter:

```
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 (&). You can specify the following values:

**&VOL**

A volume name.

**&LOC**

A volume location.

**&VOLDSN**

The file name to be written into the sequential access media labels. For example, if the applicable device class sets BKP as the tape volume prefix, a copy storage pool tape volume file name might be BKP.BFS and a database backup tape volume file name might be BKP.DBB.

**&NL**

The new line character. When you use the new line character, the command is split at the &NL variable. If required, you must specify the appropriate continuation character before the &NL character. If the &NL character 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 contains the commands that are 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 that is specified by the **SET DRMCMDFILENAME** command. 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 server.

If the operation fails after the command file is 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. Specify one of the following values:

**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. Specify one of the following values:

**No** Specifies that the server processes this command in the background.

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

To see whether 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 you continue 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 the CAP priority value is set to 0 in the library, this parameter is required. If a CAP priority value greater than 0 is set in the library, this parameter is optional. By default, all CAPs initially have a priority value of 0, which means that ACSLS does not automatically select the CAP.

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 in the range 0 - 126.
- $y$  The Library Storage Module (LSM) ID. This identifier can be a number in the range 0 - 23.
- $z$  The CAP ID. This identifier can be a number in the range 0 - 11.

For more information, see the 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

- The value of the **TOSTATE** parameter that was specified
- The next state of the **WHERESTATE** parameter that was specified, if the **TOSTATE** parameter was not specified

### Destination location

- The value of the **TOLOCATION** parameter that was specified
- The location of the **TOSTATE** parameter that was specified, if the **TOLOCATION** parameter was not specified
- The location of the next state of the **WHERESTATE** parameter that was specified, if the **TOLOCATION** and **TOSTATE** parameters are not specified

Table 189. 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
TOSTATE, WHERELOCATION	TOSTATE	Location of the TOSTATE
TOSTATE, WHERELOCATION, 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 190. State transitions for volumes

The current state of the volume	Destination state		
	MOUNTABLE	NOTMOUNTABLE	COURIER
MOUNTABLE	N	Y	Y
NOTMOUNTABLE	N	N	Y
COURIER	N	N	N
VAULT	N	N	N
VAULTRETRIEVE	N	N	N
COURIERRETRIEVE	N	N	N
ONSITERETRIEVE	N	N	N

Table 191. State transitions for volumes

The current state of the volume	Destination state	
	VAULT	VAULTRETRIEVE
MOUNTABLE	Y	N
NOTMOUNTABLE	Y	N
COURIER	Y	N
VAULT	N	N
VAULTRETRIEVE	N	N
COURIERRETRIEVE	N	N
ONSITERETRIEVE	N	N

Table 192. State transitions for volumes

The current state of the volume	Destination state	
	COURIERRETRIEVE	ONSITERETRIEVE
MOUNTABLE	N	N
NOTMOUNTABLE	N	N
COURIER	N	N
VAULT	N	N
VAULTRETRIEVE	Y	Y
COURIERRETRIEVE	N	Y
ONSITERETRIEVE	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 you change 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 VAULTRETRIEVE state

Move disaster recovery media that is in the VAULTRETRIEVE state to the ONSITERETRIEVE state. Generate a **CHECKIN LIBVOLUME** command for each volume that is successfully processed and store the commands in a file:

```

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 HSIAO 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 HSIAO 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 that was specified with the **CMDFILENAME** parameter:

```
/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

```

**Tip:** To process the **CHECKIN LIBVOLUME** commands, issue the **MACRO** command with the file name as the macro name.

## Related commands

Table 193. Commands related to MOVE DRMEDIA

Command	Description
BACKUP DB	Backs up the 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.
QUERY ACTLOG	Displays messages from the server activity log.
QUERY DRMEDIA	Displays information about disaster recovery volumes.
QUERY DRMSTATUS	Displays DRM system parameters.
QUERY PROCESS	Displays information about background processes.
SET DRMACTIVEDATASTGPOOL	Specifies that active-data storage pools are managed by DRM.

Table 193. Commands related to MOVE DRMEDIA (continued)

Command	Description
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 194. 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 is moved by 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 by using the **QUERY PROCESS** command. To cancel, issue the **CANCEL PROCESS** command.

To determine whether the command was successful, issue the **QUERY ACTLOG** command or use the server console.

The volumes that are moved by the **MOVE DRMEDIA** command for offsite recovery are not processed by the **MOVE MEDIA** command.

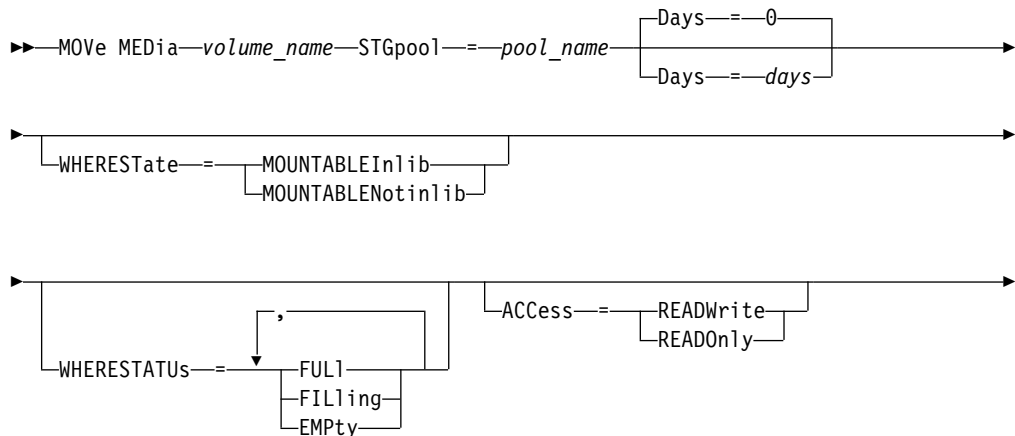
The **MOVE MEDIA** command does not process copy storage pool volumes with a DRM STATUS value of NOTMOUNTABLE, COURIER, or VAULT.

### Privilege class

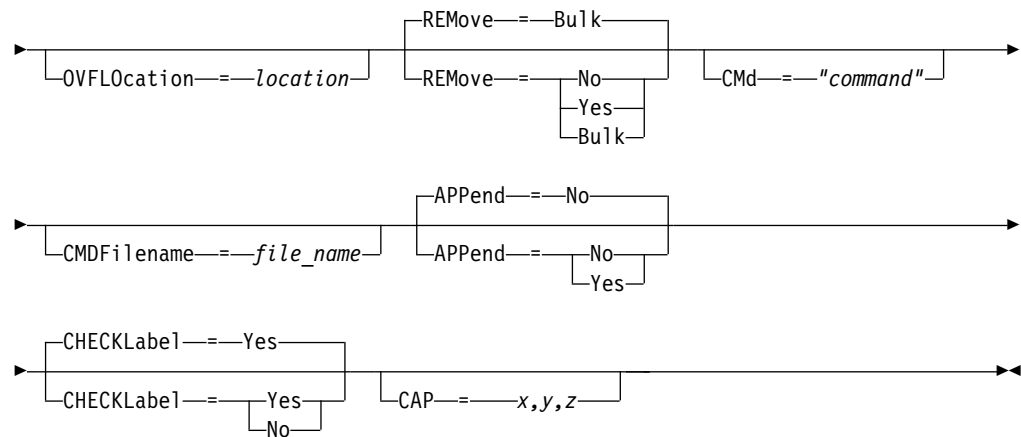
To issue this command, you must have one of the following privilege classes:

- 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* (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 is 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 move 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 might 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 might contain valid data and must be checked into the library.

#### **WHERESTATUS**

Specifies that the move process must 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 read but not write to 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.

#### **OVFL0cation**

Specifies the overflow location that is the destination of the volumes that are 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 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. The default is BULK. The response of the server to each of those options and the default values are described in the following tables.

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

Table 195. How the 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.  The server leaves the cartridge in the library in the INSERT category for use by other applications.

**SCSI libraries:** The following table shows how the server responds to YES, BULK, and NO for SCSI libraries.

Table 196. How the Server Responds for SCSI Libraries

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

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

Table 197. How the 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 <b>MOVE MEDIA</b> 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 198. How the 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, the server 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 generates a default file name by appending the string `exec.cmds.media` to the Tivoli Storage Manager server directory.

If the length of the command that is 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 must 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 might 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 treats them as strings and no substitution is 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 treats them as strings and no substitution is 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 that uses 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 does not append any continuation character. The user is responsible for specifying the correct 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 contains the commands that are 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 generates 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 that is 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 must run the commands that were 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 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 the server reads 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 YES. Possible values are:

#### Yes

Specifies that the server attempts to read the media label. Reading the media label verifies that the correct volume is being checked out.

**No** Specifies that the server 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 the CAP priority value is set to 0 in the library, this parameter is required. If a CAP priority value greater than 0 is set in the library, this parameter is optional. By default, all CAPs initially have a priority value of 0, which means that ACSLS does not automatically select the CAP.

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 in the range 0 - 126.
- $y$  The Library Storage Module (LSM) ID. This identifier can be a number in the range 0 - 23.
- $z$  The CAP ID. This identifier can be a number in the range 0 - 11.

For more information, see the 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
```

**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 199. Commands related to MOVE MEDIA*

<b>Command</b>	<b>Description</b>
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 that is in a sequential-access storage pool. You can move data for one or more nodes, a group of file spaces, or for a group of collocated nodes. You can also move selected file spaces for a single node. The data can be 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 filespaces. The syntax and parameters for each form are defined separately.

- “MOVE NODEDATA (Move data in file spaces for one or more nodes or a collocation group)” on page 644
- “MOVE NODEDATA (Move data from selected file spaces of a single node)” on page 648

**Restriction:** You cannot move node data into or out of a storage pool that is defined with a CENTERA device class.

*Table 200. 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 or file space to a collocation group.
DELETE COLLOCGROUP	Deletes a collocation group.
DELETE COLLOCMEMBER	Deletes a client node or file space 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.



*Table 200. Commands related to MOVE NODEDATA (continued)*

<b>Command</b>	<b>Description</b>
QUERY PROCESS	Displays information about background processes.
QUERY STGPOOL	Displays information about storage pools.
QUERY VOLUME	Displays information about storage pool volumes.
UPDATE COLLOGROUP	Updates the description of a collocation group.

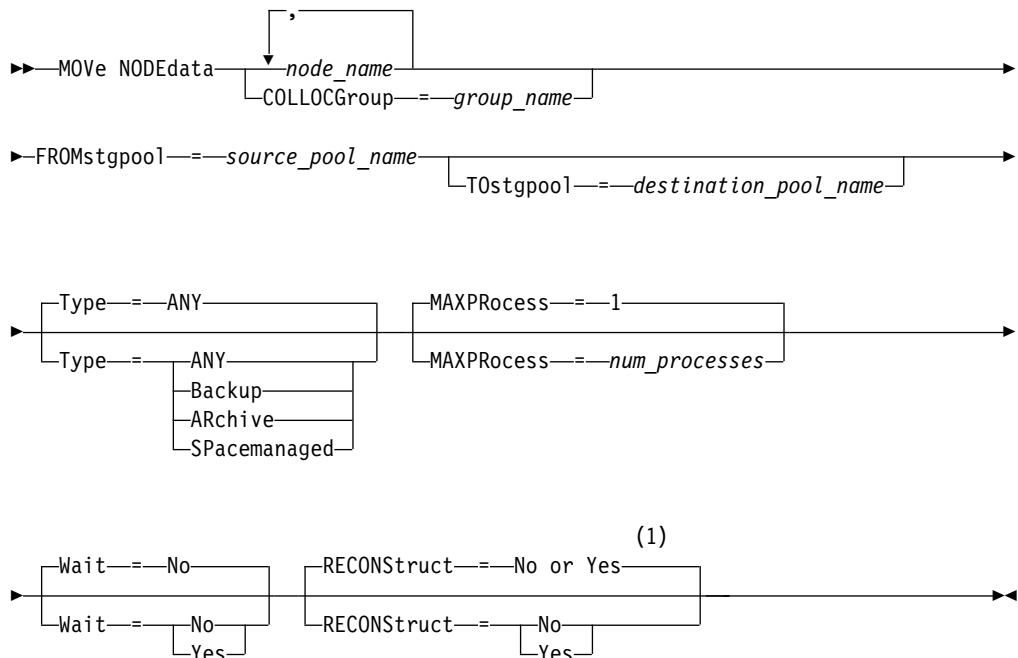
## MOVE NODEDATA (Move data in file spaces for one or more nodes or a collocation group)

Use this command to move data in file spaces that belong to; one or more nodes, a node collocation group, or a file space collocation group.

### 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 are moving data to another storage pool, you need 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 that is 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 nodes and file spaces that belong to the collocation group are 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.

**T0stgpool**

Specifies the name of a storage pool to where the data is 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 that 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. Specify one of the following values:

**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 to 999, inclusive. The default value is 1. Increasing the number of parallel processes usually improves throughput.

When you determine 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. The mount points and drives also depend 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. Specify one of the following values:

**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 might move before the cancellation.

**Yes**

Specifies that the server processes this command in the foreground. You wait for the command to complete before you can continue 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 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 any of the following conditions are true:

- The data format is NETAPPDUMP, CELERRADUMP, or NDMPDUMP.
- The data is in a storage pool that is configured for data deduplication.
- The target storage pool for the data movement is configured for data deduplication.

**Attention:** Reconstruction removes inactive backup files in active-data pools. If you specify RECONSTRUCT=NO when you move the data in an active-data pool that is not configured for data deduplication, inactive backup files remain in the storage pool.

You can specify one of the following values:

**No** Specifies that reconstruction of file aggregates are not run during the move.

**Yes**

Specifies that reconstruction of file aggregates are run during the move. You can specify only this option when both the source and the target storage pools are sequential-access.

**Move a specific node's data from a tape storage pool to a disk storage pool**

Move all data that belongs to node MARY that is stored in storage pool TAPEPOOL. Data can be moved to disk storage pool BACKUPPOOL.

```
move nodedata mary  
  fromstgpool=tapepool tostgpool=backuppoo1
```

**Move data for a node collocation group from one storage pool to another**

Move all data for node collocation group NODEGROUP1 from storage pool SOURCEPOOL to storage pool TARGETPOOL.

```
move nodedata collocgroup=nodegroup1 fromstgpool=sourcespool tostgpool=targetpool
```

## **Move data for a file space collocation group from one storage pool to another**

Move all data for file space collocation group FSGROUP1 from storage pool SOURCEPOOL2 to storage pool TARGETPOOL2.

```
move nodedata collocgroup=fsgroup1 fromstgpool=sourcespool2 tostgpool=targetpool2
```

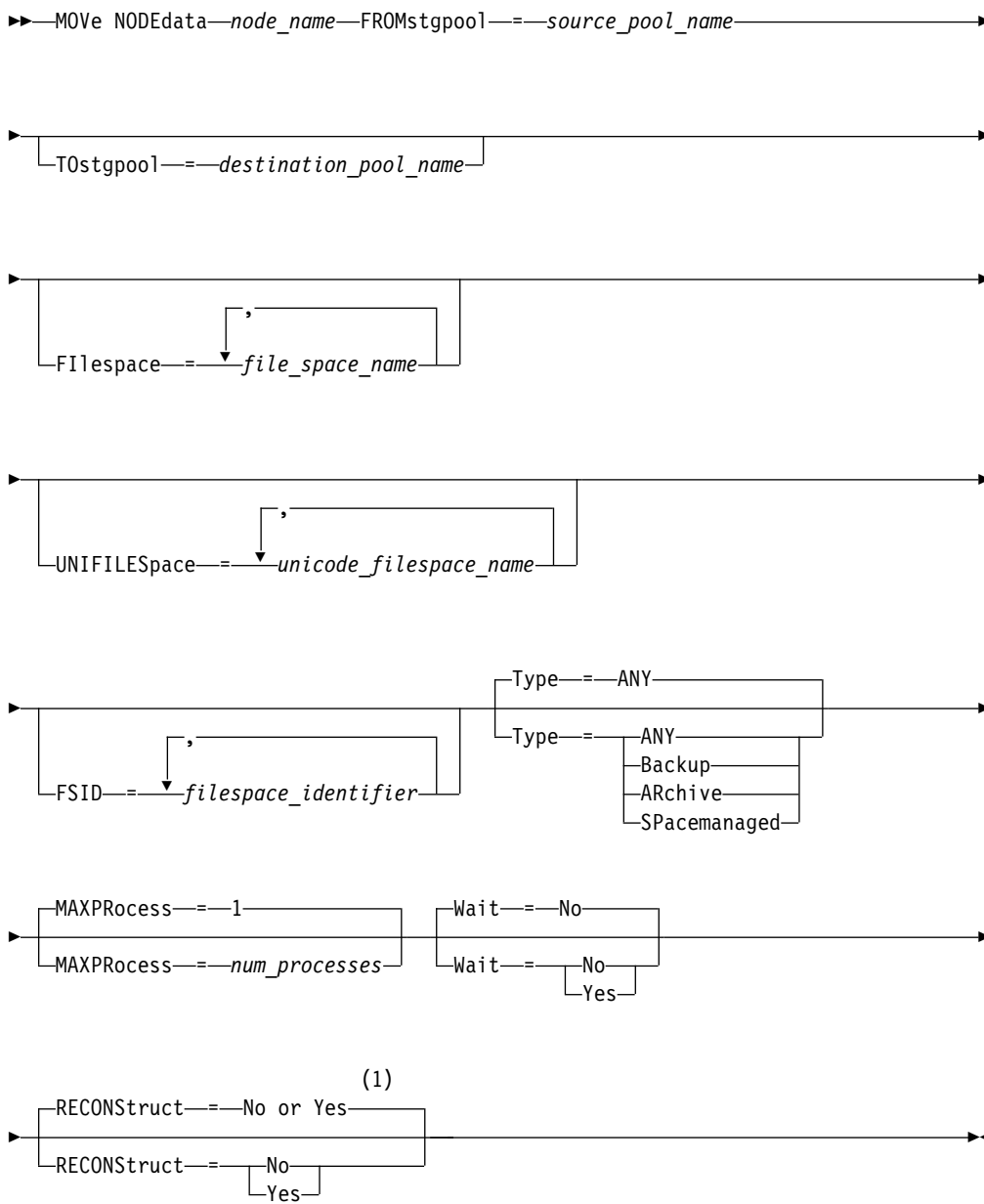
## MOVE NODEDATA (Move data from selected file spaces of a single node)

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.

**T0stgpool**

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 filesystem 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 filesystem 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.

The parameter is not available or is ignored if any of the following conditions are true:

- The data format is NETAPPDUMP, CELERRADUMP, or NDMPDUMP.



- The data is in a storage pool that is configured for data deduplication.
- The target storage pool for the data movement is configured for data deduplication.

**Attention:** Reconstruction removes inactive backup files in active-data pools. If you specify RECONSTRUCT=NO when moving the data in an active-data pool that is not configured for data deduplication, inactive backup files remain in the storage pool.

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.

### 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 filespace \\servtuc\d\$ and Unicode file space \\tsmserv1\e\$ that has a filespace 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
```

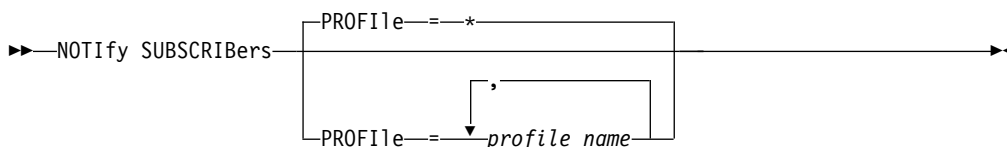
## NOTIFY SUBSCRIBERS (Notify managed servers to update profiles)

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



### 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. The default is to notify all subscribers.

### 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 201. 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.

## PERFORM LIBACTION (Define or delete all drives and paths for a library)

Use this command to define or delete all drives and their paths for a single library in one step.

This command can be used when you set up a library environment or modify an existing hardware setup that requires changes to many drive definitions. After you define a library, issue **PERFORM LIBACTION** to define drives and their paths for the library. You can also delete all drives and paths for a library by issuing the command with ACTION=DELETE.

This command is only valid for library types of SCSI and VTL. To use this command with ACTION=DEFINE, the SANDISCOVERY option must be supported and enabled.

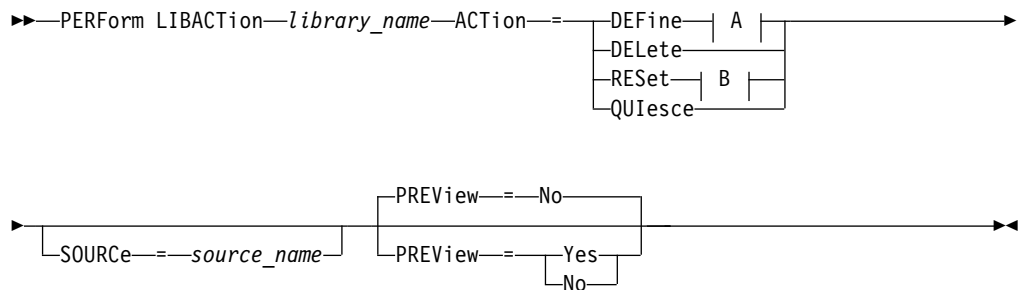
For detailed and current library support information, see the Supported Devices website for your operating system:

[http://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)

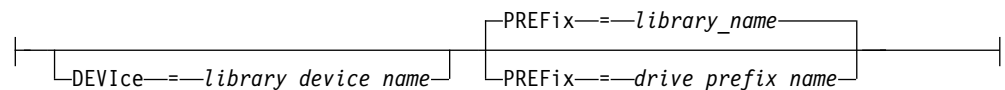
### Privilege class

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

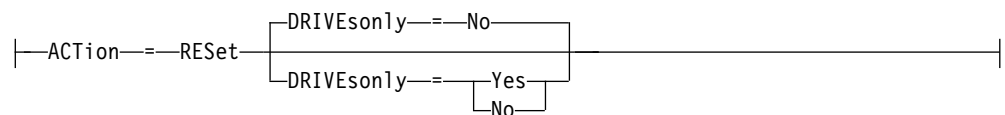
### Syntax



#### A (DEFINE):



#### B (RESET):



## Parameters

### *library\_name* (Required)

Specifies the name of the library to be defined or deleted. The maximum length of this name is 30 characters unless you are issuing **PERFORM LIBACTION** with **ACTION=DEFINE** and using the default **PREFIX** value. In that case, the maximum length of the name is 25 characters.

### **ACTion**

Specifies the action for the **PERFORM LIBACTION** command. Possible values are:

#### **DEFine**

Specifies that drives and their paths are defined for the specified library. SAN discovery must be enabled before you specify this parameter value.

#### **DELete**

Specifies that drives and their paths are deleted for the specified library.

#### **RESet**

Specifies that drives and their paths are updated online for the specified library.

#### **DRIVEsonly**

Specifies that only drives are updated online for the specified library.

Possible values are:

**No** Specifies that drives and paths are updated online.

#### **Yes**

Specifies that only drives are updated online.

#### **QUIesce**

Specifies that drives are updated offline.

### **DEVIce**

Specifies the library device name that is used when you define paths if a path to the library is not already defined. If a path is already defined, the **DEVICE** parameter is ignored. The maximum length for this value is 64 characters. This parameter is optional.

### **PREFix**

Specifies the prefix that is used for all drive definitions. For example, a **PREFIX** value of *DR* creates drives *DR0*, *DR1*, *DR2*, for as many drives as are created. If a value is not specified for the **PREFIX** parameter, the library name is used as the prefix for drive definitions. The maximum length for this value is 25 characters.

### **SOURCe**

Specifies the source server name to be used when you define or delete drive path definitions on a library client or LAN-free client. Use this parameter only if the drives in the library are set up for the local server. If no value is specified for the **SOURCE** parameter, the local server name, which is the default, is used. The maximum length for the source name is 64 characters.

If you specify the **SOURCE** parameter, you can **RESET** only paths from specified **SOURCE** values. The **SOURCE** parameter is not compatible with the **RESET**, **DRIVESONLY=YES** or **QUIESCE** options.

If a source name other than the local server name is specified with **ACTION=DEFINE**, drive path definitions are defined with the token value of **UNDISCOVERED**. The path definitions are then updated dynamically by library clients that support SAN Discovery the first time the drive is mounted.

### **PREView**

Specifies the output of all commands that are processed for **PERFORM LIBACTION** before the command is issued. The **PREVIEW** parameter is not compatible with the **DEVICE** parameter. If you are issuing the **PERFORM LIBACTION** command to define a library, you cannot specify both the **PREVIEW** and the **DEVICE** parameter.

Possible values are:

**No** Specifies that a preview of the commands that are issued for **PERFORM LIBACTION** is not displayed.

#### **Yes**

Specifies that a preview of the commands that are issued for **PERFORM LIBACTION** is displayed.

### **Example: Define a shared library**

Assume that you are working in a SAN and that you configured a library manager named LIBMGR1. Now, define a library that is named SHAREDTSM to a library client server named LIBCL1.

Issue **DEFINE LIBRARY** from the library client server, LIBCL1:

```
define library sharedtsm libtype=shared primarylibmanager=libmgr1
```

Then, issue **PERFORM LIBACTION** from the library manager, LIBMGR1, to define the drive paths for the library client:

```
perform libaction sharedtsm action=define source=libcl1
```

**Note:** The **SANDISCOVERY** option must be supported and enabled on the library client server.

### **Example: Define a library with four drives**

Define a SCSI library named KONA:

```
define library kona libtype=scsi
```

Then issue the **PERFORM LIBACTION** command to define drives and paths for the library:

```
perform libaction kona action=define device=/dev/rmt/tsmlb3  
prefix=dr
```

The server then runs the following commands:

```
define path server1 kona srct=server destt=library  
device=/dev/rmt/tsmlb3  
define drive kona dr0  
define path server1 dr0 srct=server destt=drive library=kona  
device=/dev/rmt/tsmmt1  
define drive kona dr1  
define path server1 dr1 srct=server destt=drive library=kona  
device=/dev/rmt/tsmmt2  
define drive kona dr2  
define path server1 dr2 srct=server destt=drive library=kona  
device=/dev/rmt/tsmmt3  
define drive kona dr3  
define path server1 dr3 srct=server destt=drive library=kona  
device=/dev/rmt/tsmmt4
```

## Related commands

Table 202. Commands related to **PERFORM LIBACTION**

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.
DEFINE SERVER	Defines a server for server-to-server communications.
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 LIBRARY	Changes the attributes of a library.
UPDATE PATH	Changes the attributes associated with a path.

---

## 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.

If the remote server is at the current level, the server credentials are verified automatically when you run the **PING SERVER** command. If the remote server is not at the current level, the server credentials are not verified.

### 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 203. 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 that is 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.

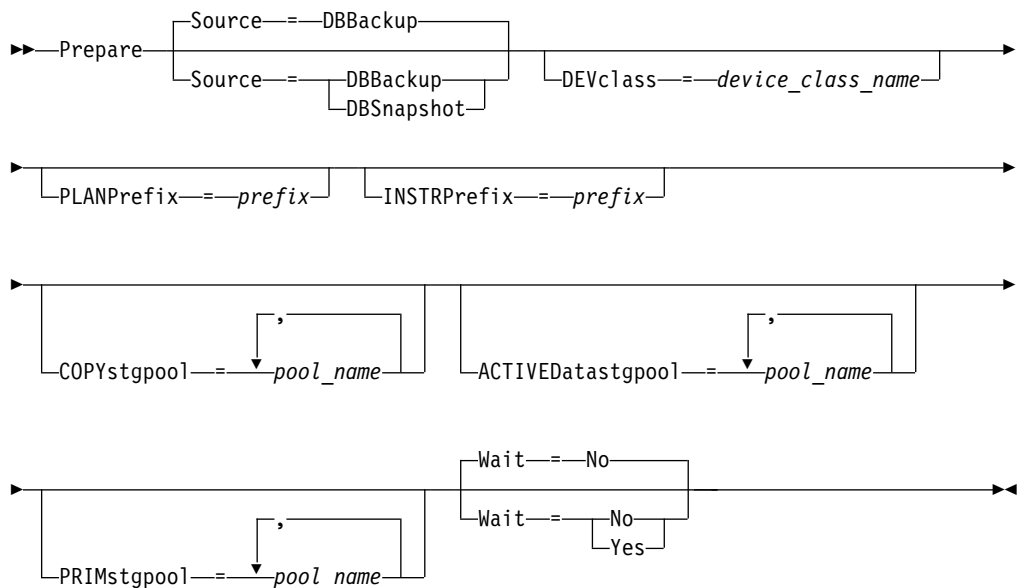
You can use the **QUERY ACTLOG** command to view whether the **PREPARE** command was successful.

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 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. This parameter is optional.

The maximum length is 250 characters.

Tivoli Storage Manager appends to the prefix the sortable date and time format yyyymmdd.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.

If the PLANPREFIX parameter is not specified, Tivoli Storage Manager selects the prefix in one of these ways:

- If the **SET DRMPPLANPREFIX** command has been issued, Tivoli Storage Manager uses the prefix specified in that command.
- If the **SET DRMPPLANPREFIX** 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.

**Important:** The recovery instructions files must be created on a file system that supports long file names.

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 /opt/tivoli/tsm/server/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
```

```
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 CSTORAGEEP.
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 204. Commands related to **PREPARE**

Command	Description
CANCEL PROCESS	Cancels a background server process.
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 RPFFILE	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 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.

*Table 204. Commands related to PREPARE (continued)*

<b>Command</b>	<b>Description</b>
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** commands to request or display information about Tivoli Storage Manager objects.

- “QUERY ACTLOG (Query the activity log)” on page 667
- “QUERY ADMIN (Display administrator information)” on page 673
- “QUERY ALERTTRIGGER (Query the list of defined alert triggers)” on page 678
- “QUERY ALERTSTATUS (Query the status of an alert)” on page 680
- “QUERY ASSOCIATION (Query client node associations with a schedule)” on page 685
- “QUERY AUDITOCUPANCY (Query client node storage utilization)” on page 687
- “QUERY BACKUPSET (Query a backup set)” on page 690
- “QUERY BACKUPSETCONTENTS (Query contents of a backup set)” on page 695
- “QUERY CLOPTSET (Query a client option set)” on page 698
- “QUERY COLLOCGROUP (Query a collocation group)” on page 700
- “QUERY CONTENT (Query the contents of a storage pool volume)” on page 703
- “QUERY COPYGROUP (Query copy groups)” on page 710
- “QUERY DATAMOVER (Display data mover definitions)” on page 715
- “QUERY DB (Display database information)” on page 718
- “QUERY DBSPACE (Display database storage space)” on page 721
- “QUERY DEVCLASS (Display information on one or more device classes)” on page 723
- “QUERY DIRSPACE (Query storage utilization of FILE directories)” on page 727
- “QUERY DOMAIN (Query a policy domain)” on page 728
- “QUERY DRIVE (Query information about a drive)” on page 731
- “QUERY DRMEDIA (Query disaster recovery media)” on page 735
- “QUERY DRMSTATUS (Query disaster recovery manager system parameters)” on page 744
- “QUERY ENABLED (Query enabled events)” on page 747
- “QUERY EVENT (Query scheduled and completed events)” on page 749
- “QUERY EVENTRULES (Query rules for server or client events)” on page 761
- “QUERY EVENTSERVER (Query the event server)” on page 764
- “QUERY EXPORT (Query for active or suspended export operations)” on page 765
- “QUERY FILESPACE (Query one or more file spaces)” on page 772
- “QUERY LIBRARY (Query a library)” on page 780
- “QUERY LIBVOLUME (Query a library volume)” on page 783
- “QUERY LICENSE (Display license information)” on page 786
- “QUERY LOG (Display information about the recovery log)” on page 790
- “QUERY MACHINE (Query machine information)” on page 792
- “QUERY MEDIA (Query sequential-access storage pool media)” on page 795
- “QUERY MGMTCLASS (Query a management class)” on page 801
- “QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804

- “QUERY MONITORSTATUS (Query the monitoring status)” on page 807
- “QUERY MOUNT (Display information on mounted sequential access volumes)” on page 811
- “QUERY NASBACKUP (Query NAS backup images)” on page 813
- “QUERY NODE (Query nodes)” on page 817
- “QUERY NODEDATA (Query client data in volumes)” on page 829
- “QUERY NODEGROUP (Query a node group)” on page 832
- “QUERY OCCUPANCY (Query client file spaces in storage pools)” on page 834
- “QUERY OPTION (Query server options)” on page 838
- “QUERY PATH (Display a path definition)” on page 840
- “QUERY POLICYSET (Query a policy set)” on page 844
- “QUERY PROCESS (Query one or more server processes)” on page 847
- “QUERY PROFILE (Query a profile)” on page 851
- “QUERY PROXYNODE (Query proxy authority for a client node)” on page 854
- “QUERY PVUESTIMATE (Display processor value unit estimate)” on page 855
- “QUERY RECOVERYMEDIA (Query recovery media)” on page 859
- “QUERY REPLICATION (Query node replication processes)” on page 862
- “QUERY REPLNODE (Display information about replication status for a client node)” on page 875
- “QUERY REPLRULE (Query replication rules)” on page 879
- “QUERY REPLSERVER (Query a replication server)” on page 881
- “QUERY REQUEST (Query one or more pending mount requests)” on page 883
- “QUERY RESTORE (Query restartable restore sessions)” on page 884
- “QUERY RPFCONTENT (Query recovery plan file contents stored on a target server)” on page 887
- “QUERY RPFFILE (Query recovery plan file information stored on a target server)” on page 889
- “QUERY SCHEDULE (Query schedules)” on page 892
- “QUERY SCRIPT (Query Tivoli Storage Manager scripts)” on page 902
- “QUERY SERVER (Query a server)” on page 905
- “QUERY SERVERGROUP (Query a server group)” on page 910
- “QUERY SESSION (Query client sessions)” on page 912
- “QUERY SHREDSTATUS (Query shredding status )” on page 917
- “QUERY SPACETRIGGER (Query the space triggers)” on page 919
- “QUERY STATUS (Query system parameters)” on page 922
- “QUERY STATUSTHRESHOLD (Query status monitoring thresholds)” on page 932
- “QUERY SSLKEYRINGPW (Query SSL key database file password)” on page 921
- “QUERY STGPOOL (Query storage pools)” on page 935
- “QUERY SUBSCRIBER (Display subscriber information)” on page 948
- “QUERY SUBSCRIPTION (Display subscription information)” on page 950
- “QUERY SYSTEM (Query the system configuration and capacity)” on page 952
- “QUERY TAPEALERTMSG (Display status of SET TAPEALERTMSG command)” on page 954
- “QUERY TOC (Display table of contents for a backup image)” on page 955

- “QUERY VIRTUALFSMAPPING (Query a virtual file space mapping)” on page 958
- “QUERY VOLHISTORY (Display sequential volume history information)” on page 960
- “QUERY VOLUME (Query storage pool volumes)” on page 968



## QUERY ACTLOG (Query the activity log)

Use this command to display messages generated by the server and client. This command provides filtering options that can be used to limit the number of messages displayed and the time that it takes to process this query. If you do not specify any parameters with this command, all messages generated in the previous hour are displayed.

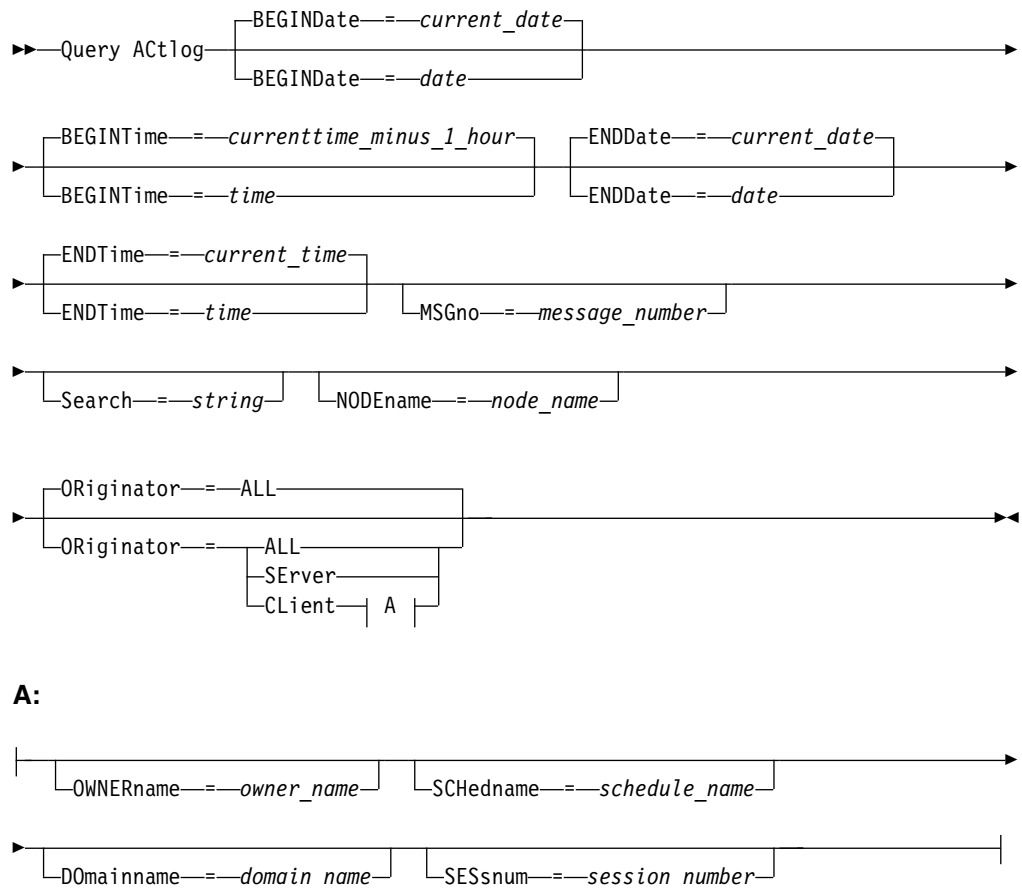
The activity log contains all messages that are sent to the server console under normal operation. The results of commands entered at the server console are not recorded in the activity log unless the command affects or starts a background process or client session. Error messages are displayed in the activity log.

**Restriction:** You cannot schedule the **QUERY ACTLOG** command by 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- <i>days</i> or <i>-days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY -7 or -7.  To display information beginning with messages created a week ago, you can specify <code>BEGINDATE=TODAY-7</code> or <code>BEGINDATE= -7</code> .
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### **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+ <i>HH:MM</i> or <i>+HH:MM</i>	The current time plus hours and minutes on the specified begin date	NOW+03:00 or +03:00.  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.

Value	Description	Example
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes on the specified begin date	NOW-04:00 or -04:00.  If you issue the QUERY ACTLOG command at 9:00 with BEGINTime=NOW-3:30 or BEGINTime= -3:30, 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
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY-1 or -1.  To display information created up to yesterday, you can specify ENDDATE=TODAY-1 or simply ENDDATE= -1.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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
<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

Value	Description	Example
NOW+HH:MM or +HH:MM	The current time plus hours and minutes on the specified end date	NOW+03:00 or +03:00.  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.  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 integer 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 includes 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).

**Example: Search activity log for messages from a 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 client and server messages from a specific client node and session

Search the activity log for Tivoli Storage Manager messages from the client and server for node A associated with Session 1. The output includes all messages with the defined text string, "SESSION: 1". Issue the command:

```
query actlog search="(SESSION:1)"
```

Date/Time	Message
02/13/2012 12:13:42	ANR0406I Session 1 started for node A (WinNT) (Tcp/Ip colind(2463)). (SESSION: 1)
02/13/2012 12:13:56	ANE4952I (ANE4985I Session: 1, ANE4986I Node: A) Total number of objects inspected: 34 (SESSION: 1)
02/13/2012 12:13:56	ANE4954I (ANE4985I Session: 1, ANE4986I Node: A) Total number of objects backed up: 34 (SESSION: 1)
02/13/2012 12:13:56	ANE4958I (ANE4985I Session: 1, ANE4986I Node: A) Total number of objects updated: 0 (SESSION: 1)
02/13/2012 12:13:56	ANE4964I (ANE4985I Session: 1, ANE4986I Node: A) Elapsed processing time: 00:00:02 (SESSION: 1)
02/13/2012 12:13:59	ANR0403I Session 1 ended for node A (WinNT). (SESSION: 1)

## Example: Search activity log for client-generated messages from a client session

Search the activity log for Tivoli Storage Manager messages from a specific client session. The output includes only messages generated by the client. Issue the command:

```
query actlog sessnum=1
```

Date/Time	Message
02/13/2012 12:13:56	ANE4952I (ANE4985I Session: 1, ANE4986I Node: A) Total number of objects inspected: 34 (SESSION: 1)
02/13/2012 12:13:56	ANE4954I (ANE4985I Session: 1, ANE4986I Node: A) Total number of objects backed up: 34 (SESSION: 1)
02/13/2012 12:13:56	ANE4958I (ANE4985I Session: 1, ANE4986I Node: A) Total number of objects updated: 0 (SESSION: 1)
02/13/2012 12:13:56	ANE4964I (ANE4985I Session: 1, ANE4986I Node: A) Elapsed processing time: 00:00:02 (SESSION: 1)

## Field descriptions

### Date/Time

Specifies the date and time when the message was generated by the server or client.

### Message

Specifies the message that was generated by the server or client.

## Related commands

Table 205. Command related to QUERY ACTLOG

Command	Description
SET ACTLOGRETENTION	Specifies the number of days to retain log records in the activity log.

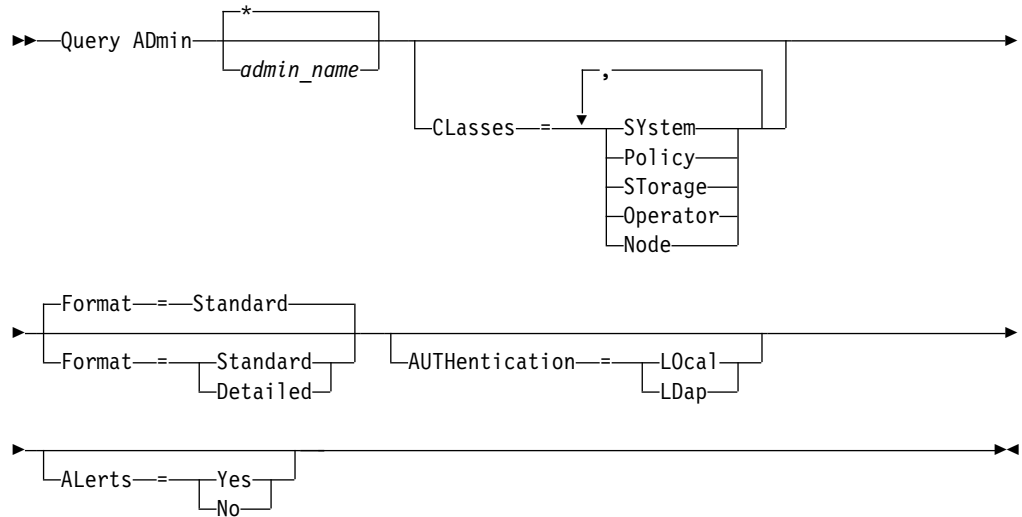
## QUERY ADMIN (Display administrator information)

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 about 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.

**Authentication**

Specifies the password authentication method for the administrator.

**Local**

Display those administrators authenticating to the IBM Tivoli Storage Manager server.

**Ldap**

Display those administrators authenticating to an LDAP directory server. The administrator password is case-sensitive.

**Alert**

Specifies whether alerts are sent to an administrators email address.

**Yes**

Specifies that alerts are sent to the specified administrators email address.

**No**

Specifies that alerts are not sent to the specified administrators email address. This is the default value.

**Tip:** Alert monitoring must be enabled, and email settings must be correctly defined to successfully receive alerts by email. To view the current settings, issue the **QUERY MONITORSETTINGS** command.

**Example: Display information about all administrators**

Display partial information on all administrators. Issue the command:

```
query admin
```

Administrator Name	Days Since Last Access	Days Since Password Set	Locked?	Privilege Classes
ADMIN	<1	<1	No	System
SERVER_CONSOLE			No	System

See "Field descriptions" on page 675 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:
  Email Aerts: Yes
  Authentication: Local
  SSL Required: No
```

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**

Specifies the email address for the administrator.

**Email Alerts**

Specifies whether alerts are sent to the specified administrator by email.

**Authentication**

Specifies the password authentication method: LOCAL, LDAP, or LDAP (pending).

Authentication Target	Authentication Method
Tivoli Storage Manager server	LOCAL
LDAP directory server	LDAP
This administrator is configured to authenticate with an LDAP directory server, but the administrator did not yet authenticate through a client node.	LDAP (pending)

**SSL Required**

Specifies if the security setting for the administrator user ID requires Secure Sockets Layer (SSL). Values can be YES, NO, or Default. You must have system level authority to update the administrator **SSLREQUIRED** setting.

## Related commands

Table 206. 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 a 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.

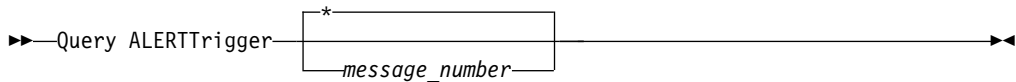
## QUERY ALERTTRIGGER (Query the list of defined alert triggers)

Use this command to display which server messages are defined as alerts.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *message\_number*

Specifies the message number that you want to query. Specify multiple message numbers, which are separated by commas, and no intervening spaces. Message numbers are a maximum of eight characters in length. Wildcard characters can be used to specify message numbers. If you do not specify a message number, all alert triggers are displayed.

### Query alert triggers to display which messages are designated as alerts

Display all messages that are designated as alerts by issuing the following command:

```
query alerttrigger
```

Example output:

Alert Trigger	Category	Administrator
ANR1067E	SERVER	HARRYH
ANR1073E	SERVER	CSDADMIN, DJADMIN, HARRYH
ANR1074E	STORAGE	CSDADMIN, DJADMIN, HARRYH
ANR1096E	STORAGE	CSDADMIN, DJADMIN, HARRYH, MHAYE

### Query alert triggers for a specific message number

Display all alert triggers that have message number ANR1067E designated to them by issuing the following command:

```
query alerttrigger ANR1067E
```

Example output:

Alert Trigger	Category	Administrator
ANR1067E	SERVER	HARRYH

### Related commands

Table 207. Commands related to QUERY ALERTTRIGGER

Command	Description
“DEFINE ALERTTRIGGER (Define an alert trigger)” on page 124	Associates specified messages to an alert trigger.

Table 207. Commands related to **QUERY ALERTTRIGGER** (continued)

<b>Command</b>	<b>Description</b>
"DELETE ALERTTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.
"QUERY ALERTSTATUS (Query the status of an alert)" on page 680	Displays information about alerts that have been issued on the server.
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.
"UPDATE ALERTSTATUS (Update the status of an alert)" on page 1205	Updates the status of a reported alert.

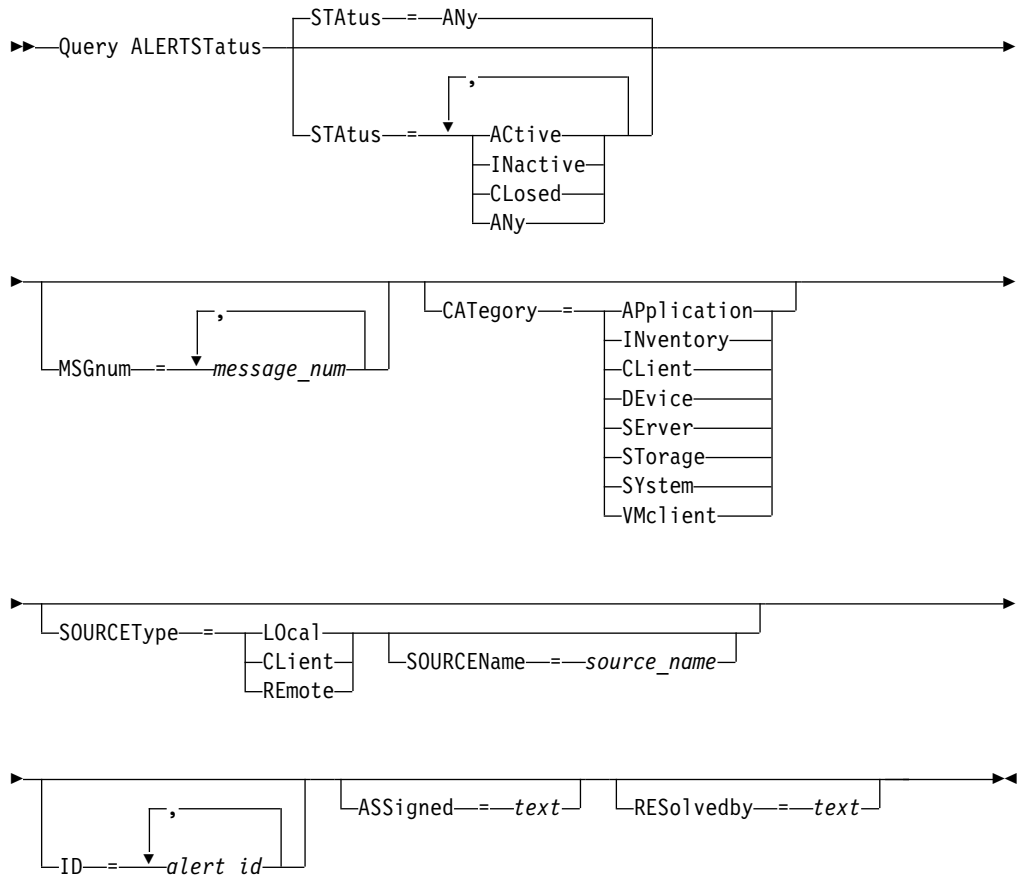
## QUERY ALERTSTATUS (Query the status of an alert)

Use this command to display information about alerts that are reported on the Tivoli Storage Manager server.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### Status

Specifies the status type that you want to display. If you do not specify a status, all alerts are queried and displayed. Specify one of the following values:

##### Active

Displays alerts that are specified in the Tivoli Storage Manager server database as active.

##### InActive

Displays alerts that are in the inactive state.

##### Closed

Displays alerts that are in the closed state.

**ANy**

Displays all alerts, without regard to state.

**MSGnum**

Specifies the message number that you want to display. Specify the numerical portion of a Tivoli Storage Manager server message. Values are in the range 0 - 9999. For example, the message number in message ANR2044E is 2044. Specify multiple message numbers by separating them with commas and no intervening spaces.

**CATegory**

Specifies the category type for the alert, which is determined by the message types. Specify one of the following values:

**APplication**

Alert is classified as application category. For example, you can specify this category for messages that are associated with application (TDP) clients.

**INventory**

Alert is classified as inventory category. For example, you can specify this category for messages that are associated with the database, active log file, or archive log file.

**Note:** The category of CAAtalog is used instead of INventory in alerts from servers that were not upgraded to Tivoli Storage Manager 7.1.0 or later.

**CLient**

Alert is classified as client category. For example, you can specify this category for messages that are associated with general client activities.

**DEvice**

Alert is classified as device category. For example, you can specify this category for messages that are associated with device classes, libraries, drives, or paths.

**SErver**

Alert is classified as general server category. For example, you can specify this category for messages that are associated with general server activities or events.

**STorage**

Alert is classified as storage category. For example, you can specify this category for messages that are associated with storage pools.

**SYstems**

Alert is classified under system clients category. For example, you can specify this category for messages that are associated with system backup and archive or hierarchical storage management (HSM) backup-archive clients.

**VMclient**

Alert is classified under VMclient category. For example, you can specify this category for messages that are associated with virtual machine clients.

**SOURCEType**

Specifies the source type that is being queried. Specify one of the following values:

**LOcal**

Displays alerts that originated from the local Tivoli Storage Manager server.

**Client**

Displays alerts that originated from the Tivoli Storage Manager client.

**REmote**

Displays alerts that originated from another Tivoli Storage Manager server.

**SOURCENAME**

Specifies the name of the source where the alert originated. **SOURCENAME** can be the name of a local or remote Tivoli Storage Manager server, or a Tivoli Storage Manager client.

**ID** This optional parameter specifies the unique ID of the alert that you want to display. Specify a value from 1 to 9223372036854775807.

**ASSigned**

Specifies the administrator name that is assigned the alert that you want to query.

**RESolvedby**

Specifies the administrator name that resolved the alert that you want to query.

**Query active alerts**

Display only alerts that are active in the server database by issuing the following command:

```
query alertstatus status=active
```

**Query active alerts for two messages issued by the local server**

Issue the following command to display only active alerts for message numbers ANE4958I and ANR4952E that were issued by the local server:

```
query alertstatus msgnum=4958,4952 status=active sourcetype=local
```

**Query active alerts for messages ANR4958I and ANR4952E issued by a client**

Issue the following command to display only active alerts for message numbers ANE4958I and ANE4952I that were issued by a client:

```
query alertstatus msgnum=4958,4952 status=active sourcetype=client
```

**Query all alerts on a server**

Issue the following command to display all alerts that are on the server:

```
query alertstatus
```

Example output: Display all the alerts that are on the server:



```

Alert Identifier: 83
Alert Message Number: 293
  Source Name: SEDONA
  Source Type: LOCAL
  First Occurrence: 03/07/2013 17:08:35
Most Recent Occurrence: 03/07/2013 17:08:35
  Count: 1
  Status: ACTIVE
  Last Status Change: 12/31/1969 17:00:00
  Category: INVENTORY
  Message: ANR0293I Reorganization for table AF_BITFILES started.
  Assigned:
  Resolved By:
  Remark:

Alert Identifier: 85
Alert Message Number: 293
  Source Name: SEDONA
  Source Type: LOCAL
  First Occurrence: 03/08/2013 05:45:00
Most Recent Occurrence: 03/08/2013 05:45:00
  Count: 1
  Status: ACTIVE
  Last Status Change: 12/31/1969 17:00:00
  Category: INVENTORY
  Message: ANR0293I Reorganization for table BF_AGGREGATED_BITFILES started.
  Assigned:
  Resolved By:
  Remark:

Alert Identifier: 1282
Alert Message Number: 293
  Source Name: ALPINE
  Source Type: LOCAL
  First Occurrence: 02/13/2013 15:47:50
Most Recent Occurrence: 02/13/2013 15:47:50
  Count: 1
  Status: CLOSED
  Last Status Change: 02/26/2013 09:46:39
  Category: INVENTORY
  Message: ANR0293I Reorganization for table TSMMON_ALERT started.
  Assigned:
  Resolved By:
  Remark:

Alert Identifier: 1792
Alert Message Number: 293
  Source Name: ALPINE
  Source Type: LOCAL
  First Occurrence: 02/19/2013 08:58:14
Most Recent Occurrence: 02/19/2013 08:58:14
  Count: 1
  Status: CLOSED
  Last Status Change: 03/01/2013 12:39:21
  Category: INVENTORY
  Message: ANR0293I Reorganization for table ACTIVITY_LOG started.
  Assigned:
  Resolved By:
  Remark:

```

## Related commands

*Table 208. Commands related to QUERY ALERTSTATUS*

Command	Description
"DEFINE ALERTRIGGER (Define an alert trigger)" on page 124	Associates specified messages to an alert trigger.

Table 208. Commands related to **QUERY ALERTSTATUS** (continued)

Command	Description
"DELETE ALERTTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.
"UPDATE ALERTSTATUS (Update the status of an alert)" on page 1205	Updates the status of a reported alert.

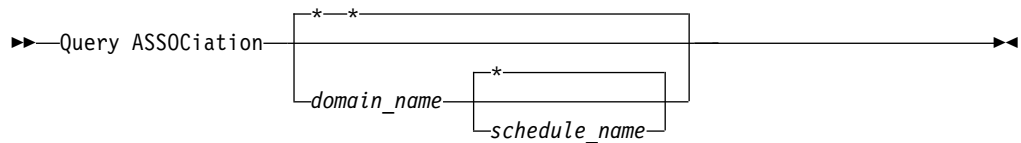
## QUERY ASSOCIATION (Query client node associations with a schedule)

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 209. Commands related to QUERY ASSOCIATION*

<b>Command</b>	<b>Description</b>
DEFINE ASSOCIATION	Associates clients with a schedule.
DELETE ASSOCIATION	Deletes the association between clients and a schedule.



**PRimary**

Specifies primary storage pools only.

**C0py**

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.

Node Name	Backup Storage Used (MB)	Archive Storage Used (MB)	Space-Managed Storage Used (MB)	Total Storage Used (MB)
CLIENT	245	20	0	265
SMITH	245	20	0	265
SMITHERS	245	20	0	265
JOHNSON	300	15	0	320
JOE	245	20	0	265
TOM	300	15	0	320
LARRY	245	20	0	265

See “Field descriptions” for field descriptions.

**Field descriptions****Node Name**

Specifies the name of the client node.

**Backup Storage Used (MB)**

Specifies the total backup storage use for the node. For this value, one MB = 1048576 bytes.

**Archive Storage Used (MB)**

Specifies the total archive storage use for the node. For this value, one MB = 1048576 bytes.

**Space-Managed Storage Used (MB)**

Specifies the amount of server storage that is used to store files that are migrated from the client node by a Tivoli Storage Manager for Space Management client. For this value, one MB = 1048576 bytes.

**Total Storage Used (MB)**

Specifies the total storage use for the node. For this value, one MB = 1048576 bytes.

**Related commands**

Table 210. Commands related to QUERY AUDITOCCUPANCY

Command	Description
AUDIT LICENSES	Verifies 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.

Table 210. Commands related to QUERY AUDIT OCCUPANCY (continued)

Command	Description
REGISTER LICENSE	Registers a license with the Tivoli Storage Manager server.
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.





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 + <i>days</i>	The current date plus days specified.	TODAY +3 or +3.
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### **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
NOW	The current time	NOW
NOW+ <i>HH:MM</i> or + <i>HH:MM</i>	The current time plus hours and minutes specified	NOW+02:00 or +02:00.
NOW- <i>HH:MM</i> or - <i>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
TODAY	The current date	TODAY
TODAY+ <i>days</i> or + <i>days</i>	The current date plus days specified. The maximum number of days you can specify is 9999.	TODAY +3 or +3.
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### 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> or + <i>HH:MM</i>	The current time plus hours and minutes specified	NOW+02:00 or +02:00.
NOW- <i>HH:MM</i> 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.

**WHEREToCexists**

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 a 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.

**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 211. 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)

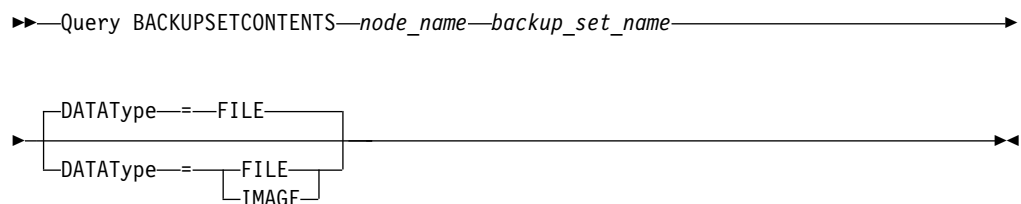
Use this command to display information about the files and directories contained in a backup set for a client node.

**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



### 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.

### 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	/dsmxxxx.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 can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

### 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 in the Operations 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.

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 212. Commands related to `QUERY BACKUPSETCONTENTS`

Command	Description
<code>DEFINE BACKUPSET</code>	Defines a previously generated backup set to a server.
<code>GENERATE BACKUPSET</code>	Generates a backup set of a client's data.
<code>GENERATE BACKUPSETTOC</code>	Generates a table of contents for a backup set.
<code>DELETE BACKUPSET</code>	Deletes a backup set.
<code>QUERY BACKUPSET</code>	Displays backup sets.
<code>UPDATE BACKUPSET</code>	Updates a retention value associated with a backup set.

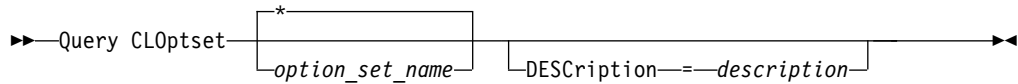
## QUERY CLOPTSET (Query a client option set)

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:
Replica Option Set: Yes

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.



**Managing profile**

Specifies the profile to which the managed server subscribed to get the definition of the client option set.

**Replica Option Set**

Specifies the replica option set is replicated by the source replication server.

**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 option setting for the client. NO indicates that the server option setting does not override the client option. YES indicates that the server 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 213. Commands related to QUERY CLOPTSET*

<b>Command</b>	<b>Description</b>
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.

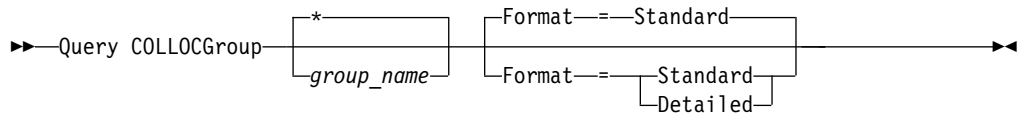
## QUERY COLLOCGROUP (Query a collocation group)

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.

### 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 701 for field descriptions.

### 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
```

```

Collocation Group Name: DEPT_ED
Collocation Group Description: Education department
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2013 10:59:03
Collocation Group Member(s): EDU_1 EDU_7
Filespace Member(s):

```

```

Collocation Group Name: GROUP1
Collocation Group Description: Low cap client nodes.
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2013 10:59:16
Collocation Group Member(s): CHESTER
Filespace Member(s): alpha

```

```

Collocation Group Name: GROUP1
Collocation Group Description: Low cap client nodes.
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2013 10:59:16
Collocation Group Member(s): CHESTER
Filespace Member(s): beta

```

```

Collocation Group Name: GROUP1
Collocation Group Description: Low cap client nodes.
Last Update by (administrator): SERVER_CONSOLE
Last Update Date/Time: 04/21/2013 10:59:16
Collocation Group Member(s): CHESTER
Filespace Member(s): gamma

```

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.

### Filespace Member(s)

The file space or file spaces that are members of the collocation group. If there is more than one file space, each file space is displayed in a separate entry.

## Related commands

Table 214. Commands related to QUERY COLLOGROUP

Command	Description
DEFINE COLLOGROUP	Defines a collocation group.
DEFINE COLLOCMEMBER	Adds a client node or file space to a collocation group.
DEFINE STGPOOL	Defines a storage pool as a named collection of server storage media.

Table 214. Commands related to QUERY COLLOGROUP (continued)

<b>Command</b>	<b>Description</b>
DELETE COLLOGROUP	Deletes a collocation group.
DELETE COLLOCMEMBER	Deletes a client node or file space 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 COLLOGROUP	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, and the names of client files that link to a deduplicated group of files.

You can use this command to identify files that the server found to be damaged and files that were 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

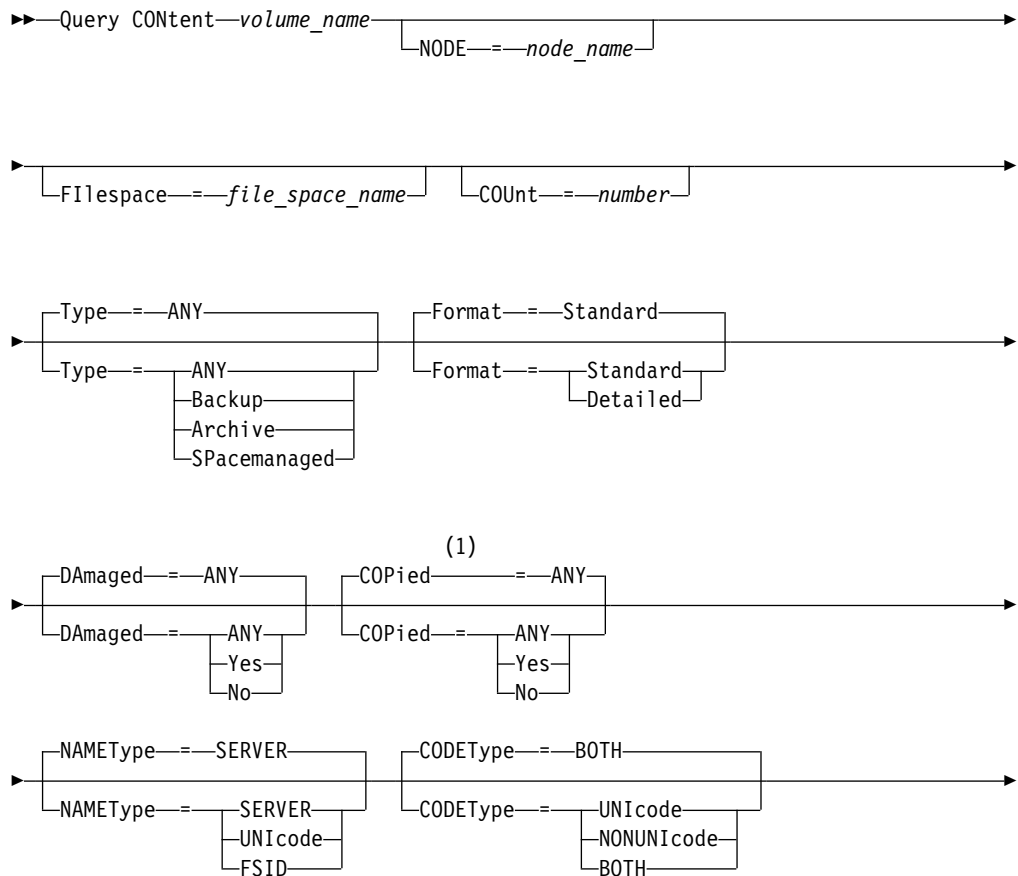
Because this command can take a long time to run and the results can be large, consider using the **COUNT** parameter to limit the number of files displayed.

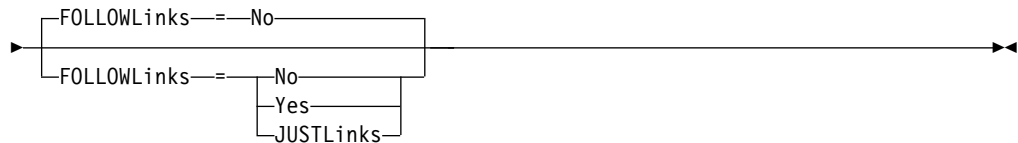
**Note:** Files that are cached in a disk volume and that are marked as damaged are not included in the results.

### 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 might need to have the server convert the file space name that you enter. For example, you might need to have the server convert the name that 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 that is 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 that are 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 are marked as damaged. For purposes of this criteria, the server examines only physical files (a file that might 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 found the files to be damaged.

**Yes**

Specifies that only files that are 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 be damaged are displayed.

**COPied**

Specifies criteria to restrict the query output based on whether files were backed up to a copy storage pool. Whether files are stored in an active-data pool does not affect 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 file spaces that are only in Unicode.

**NONUNICODE**

Include file spaces that are not only in Unicode.

**BOTH**

Include file spaces regardless of code page type.

**FOLLOWLinks**

Specifies whether to display only the files that are stored on the volume or only files that are linked to the volume. You can also display both stored files and linked files. The default is NO. Possible values are:

**No** Display only the files that are stored in the volume. Do not display files that have links to the volume.

**Yes**

Display all files, including files that are stored on the volume and any files that have links to the volume.

**JUSTLinks**

Display only the files that have links to the volume. Do not display files that are stored on the volume.

**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 Name	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 708 for field descriptions.

### Example: Display detailed information for a tape volume

Query the contents of the tape volume named WPD001. Display only files that are backed up by the node MARK, and files that are either stored on the volume or linked to the volume. Display only the first four files on the volume.

```
query content wpd001 node=mark count=4 type=backup followlinks=yes
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
Linked: No
Fragment Number:

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
Linked: No
Fragment Number: 2

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
Linked: No
Fragment Number: 3

```

See “Field descriptions” on page 708 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 can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

### 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 in the Operations 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 might display with a combination of invalid characters or blank spaces. The results of the conversion for characters that are 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.

### Hexadecimal Client's Name for File

The client's name for the file that is 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 displays "no".

**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.

**Linked**

Indicates whether the file is stored on the volume or whether the file is linked to the volume.

**Fragment Number**

Specifies the fragment number. If the fragment number is blank, it is either the first fragment or not a fragment.

**Related commands**

*Table 215. Commands related to QUERY CONTENT*

<b>Command</b>	<b>Description</b>
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.
COPY ACTIVATEDATA	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.

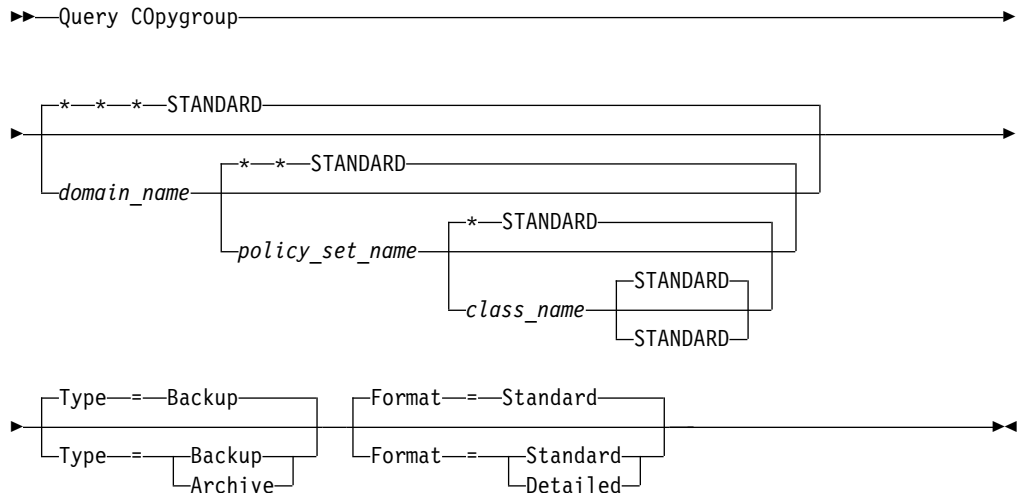
## QUERY COPYGROUP (Query copy groups)

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 that is 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 that is 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.

**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 information about the default backup copy group**

Display information about the default backup copy group in the ENGPOLDOM engineering policy domain. Issue the following command:

```
query copygroup engpoldom * *
```

The following data shows the output from the query. It shows that the ACTIVE policy set contains two backup copy groups that belong to the MCENG and STANDARD management classes.

Policy Domain Name	Policy Set Name	Mgmt Class Name	Copy Group Name	Versions Data Exists	Versions Data Deleted	Retain Extra Versions	Retain Only Version
ENGPOLDOM	ACTIVE	MCENG	STANDARD	5	4	90	600
ENGPOLDOM	ACTIVE	STANDARD	STANDARD	2	1	30	60
ENGPOLDOM	STANDARD	MCENG	STANDARD	5	4	90	600
ENGPOLDOM	STANDARD	STANDARD	STANDARD	2	1	30	60
ENGPOLDOM	TEST	STANDARD	STANDARD	2	1	30	60

**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
```

**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 Version

The number of days to retain the last backup version of a file that has been deleted from the client file system.

### 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 216. Commands related to QUERY COPYGROUP*

<b>Command</b>	<b>Description</b>
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.



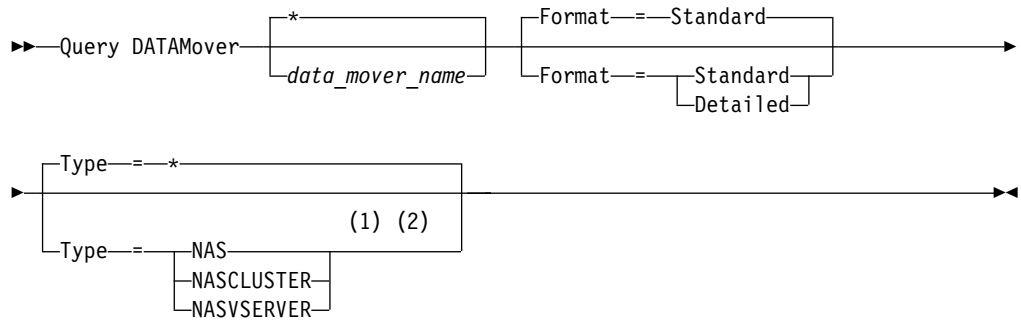
## QUERY DATAMOVER (Display data mover definitions)

Use this command to display data mover definitions.

### Privilege class

Any administrator can issue this command.

### Syntax



### Notes:

- 1 You must specify the TYPE parameter if FORMAT=DETAILED.
- 2 You can specify TYPE=NASCLUSTER and TYPE=NASVSERVER only on an AIX, Linux, Solaris, or Windows operating system.

### 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.

#### Standard

Specifies that name and address information is displayed.

#### Detailed

Specifies that complete information is displayed.

#### Type

Specifies the type of data mover to be displayed. If you specify FORMAT=DETAILED, you must specify a value for the TYPE parameter.

#### NAS

Specifies a NAS file server.

### Example: Display information about all data movers

Display the data movers on the server. Issue the command:

```
query datamover
```

Data Mover Name	Data Mover Type	Online
NASMOVER1	NAS	Yes
NASMOVER2	NAS	No

See “Field descriptions” for field descriptions.

### Example: Display information about one data mover

Display partial information about 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 about one data mover

Display detailed information about 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
Data Mover Type:	NAS
IP Address:	198.51.100.0
TCP/IP Port Number:	10000
User Name:	NDMPadmin
Storage Pool Data Format:	NDMPDUMP
Online:	Yes
Last Update by (administrator):	ADMIN
Last Update Date/Time:	05/23/2015 09:26:33

See “Field descriptions” for field descriptions.

## Field descriptions

### Data Mover Name

Specifies the name of the data mover.

### Data Mover Type

Specifies the type of the data mover.

### IP Address

Specifies the IP address of the data mover.

### TCP/IP Port Number

Specifies the TCP port number for the data mover.

### User Name

Specifies the user ID that the server uses to access the data mover.

### Storage Pool Data Format

Specifies the data format that is used by the data mover.

### Online

Specifies whether the data mover is online and available for use.

### Last Update by (administrator)

Specifies the ID of the administrator who completed the last update.

### Last Update Date/Time

Specifies the date and time when the last update occurred.

## Related commands

*Table 217. Commands related to QUERY DATAMOVER*

<b>Command</b>	<b>Description</b>
DEFINE DATAMOVER	Defines a data mover to the Tivoli Storage Manager server.
DELETE DATAMOVER	Deletes a data mover.
UPDATE DATAMOVER	Changes the definition for a data mover.



```

Database Name: TSM_DB2
Total Space of File System (MB): 1,748,800
Space Used on File System (MB): 2,304,355
Space Used by Database (MB): 448
Free Space Available (MB): 235,609
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
Package Cache Hit Ratio: 89.8
Last Database Reorganization: 05/25/2009 16:44:06
Full Device Class Name: FILE
Number of Database Backup Streams: 4
Incrementals Since Last Full: 0
Last Complete Backup Date/Time: 05/18/2009 22:55:19
Compress Database Backups: Yes
Protect Master Encryption Key: No

```

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 Space of File System (MB)

The total space, in megabytes, of the file systems in which the database is located.

### Space Used on File System (MB)

The amount of database space, in megabytes, that is in use.

### Space Used by Database (MB)

The size of the database, in megabytes. The value does not include any temporary table space. The size of the database is calculated from the amount of space that is used on the file system containing the database.

### Free Space Available (MB)

The amount of database space, in megabytes, that is not in use.

### 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 total number of free pages in all table spaces. The Tivoli Storage Manager database has up to 10 table spaces.

### 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.

**Package Cache Hit Ratio**

A percentage that indicates 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 that it is successful in avoiding these activities.

**Last Database Reorganization**

The last time that the database manager completed an automatic reorganization activity.

**Full Device Class Name**

The name of the device class that is used for full database backups.

**Number of Database Backup Streams**

The number of concurrent data movement streams that were used during the database backup.

**Incrementals Since Last Full**

The number of incremental backups that were completed since the last full backup.

**Last Complete Backup Date/Time**

The date and time of the last full backup.

**Compress Database Backups**

Specifies whether database backups are compressed.

**Protect Master Encryption Key**

Specifies whether database backups include a copy of the server master encryption key.

**Related commands**

*Table 218. Commands related to QUERY DB*

<b>Command</b>	<b>Description</b>
BACKUP DB	Backs up the 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 Space of File System (MB)	Used Space on File System (MB)	Free Space 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 Space of File System (MB)

The total amount of space, in megabytes, of the file system in which the database is located.

#### Used Space on File System (MB)

The amount of storage space, in megabytes, that is in use.

When you run the **QUERY DBSPACE** command, the value in the output might be greater than the value that is obtained by running the `df` system command. The output from the `df` system command does not include the amount of space that is reserved for the root user.

#### Free Space Available (MB)

The amount of space, in megabytes, that is not in use.

### Related commands

Table 219. Commands related to QUERY DBSPACE

Command	Description
BACKUP DB	Backs up the Tivoli Storage Manager database to sequential access volumes.

*Table 219. Commands related to QUERY DBSPACE (continued)*

<b>Command</b>	<b>Description</b>
EXTEND DBSPACE	Adds directories to increase space for use by the database.
QUERY DB	Displays allocation information about the database.



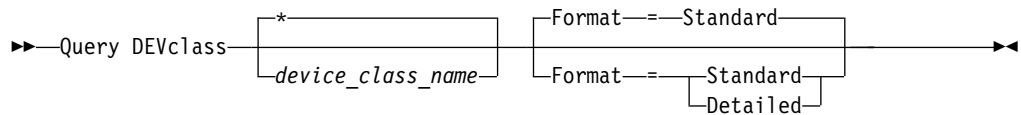
## QUERY DEVCLASS (Display information on one or more device classes)

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
DISK	Random	3				
8MMTAPE	Sequential	1	GENTAPE	DRIVE	6,144.0	2
PLAINFILES	Sequential	1	FILE		50.0	1

See “Field descriptions” on page 724 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
```

```

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:

Logical Block Protection:
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 full details 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:

High-level Address:
  WORM: No
  Scaled Capacity: 90
  Drive Encryption: On

Logical Block Protection: Read/Write
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 that is 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 that are 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 attempts to contact a target server if communications failure is suspected.

**Retry Interval**

How often the retries are done within a retry period.

**Shared**

Whether this FILE device class is shared between the server and one or more storage agents.

**High-level Address**

The IP address of the device in dotted decimal format.

**Minimum Capacity**

The minimum capacity of a volume that is associated with the device class.

**WORM**

Whether this drive is a write once, read many (WORM) device.

**Drive Encryption**

Whether drive encryption is allowed. This field applies only to volumes in a storage pool that is associated with a device type of 3592, LTO, or ECARTRIDGE.

**Scaled Capacity**

The percentage of the media capacity that can be used to store data.

**Logical Block Protection**

Specifies whether logical block protection is enabled and, if it is, the mode. Possible values are Read/Write, Write-only, and No. You can use logical block protection only with the following types of drives and media:

- IBM LTO5 and later
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later
- Oracle StorageTek T10000C and T10000D drives

**Last Update by (administrator)**

The administrator that made the last update to the device class.

**Last Update Date/Time**

The date and time of the last update.

**Related commands**

*Table 220. Commands related to QUERY DEVCLASS*

<b>Command</b>	<b>Description</b>
DEFINE DEVCLASS	Defines a device class.
DEFINE SERVER	Defines a server for server-to-server communications.
DELETE DEVCLASS	Deletes a device class.
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 221. Commands related to QUERY DIRSPACE

Command	Description
DEFINE DEVCLASS	Defines a device class.
DELETE DEVCLASS	Deletes a device class.
QUERY DEVCLASS	Displays information about device classes.
UPDATE DEVCLASS	Changes the attributes of a device class.

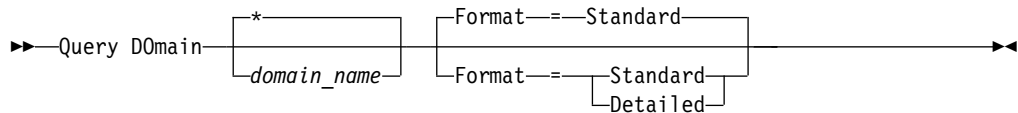
## QUERY DOMAIN (Query a policy domain)

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 729 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.

**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 222. Commands related to QUERY DOMAIN*

<b>Command</b>	<b>Description</b>
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.



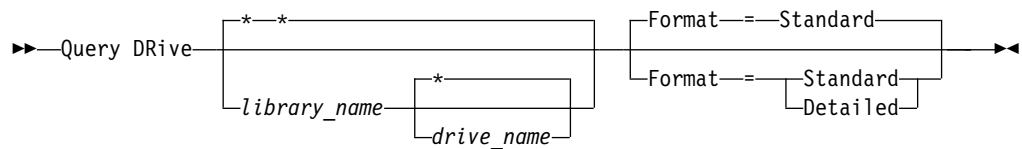
## QUERY DRIVE (Query information about a drive)

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 732 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 lib2 drive02 format=detailed
```

```
Library Name: LIB2
Drive Name: DRIVE02
Device Type: 3590
On-Line: Yes
Drive State: Empty
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 polls a drive before stating it is unavailable. The time output follows the format: mm/dd/yy hh:mm:ss.

**Read Formats**

The read formats for the drive.

**Write Formats**

The write formats for the drive.

**Element**

The element number for the drive.

**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.

**Volume Name**

The volume name for the drive.

**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.

**WWN** The World Wide Name for the drive.

**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 (Gigabytes/ASNEEDED/NONE)**

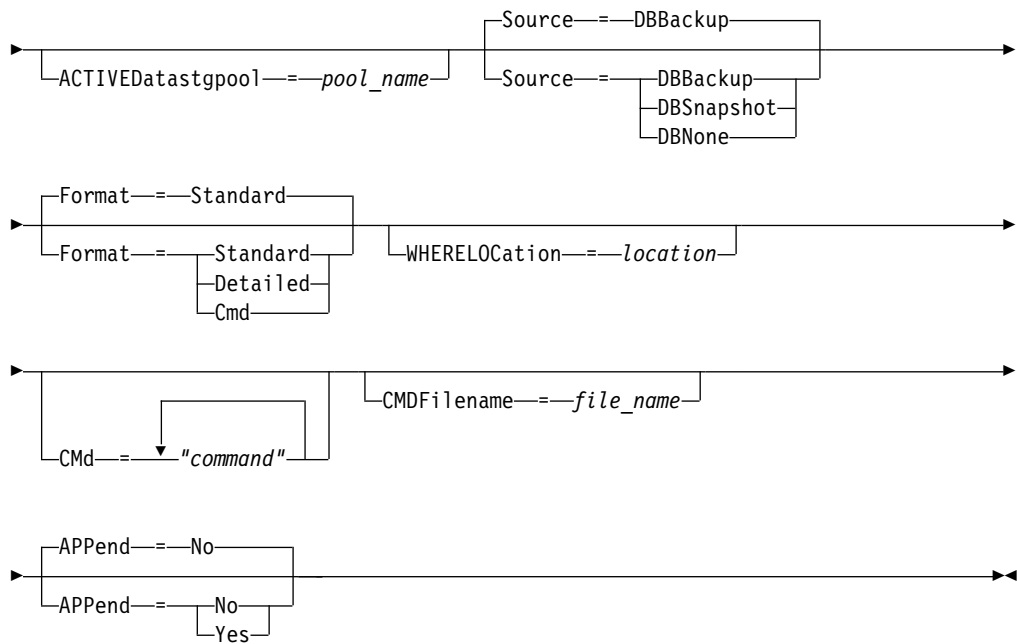
How often the server activates drive cleaning. This value can be the number of gigabytes, ASNEEDED, or NONE.

## Related commands

Table 223. 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.
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.





## Parameters

### *volume\_name*

Specifies the names of the database backup and copy storage poolvolumes 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 specified by the **COPYSTGPOOL** parameter. If you do not use the **COPYSTGPOOL** parameter, the server queries volumes from copy storage pools previously specified 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.

#### **MOnstable**

Volumes in this state contain valid data and are accessible for onsite processing.

#### **NOTMOnstable**

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.  To query volumes beginning with records changed to their current state a week ago, you can specify <b>BEGINDATE=TODAY-7</b> or <b>BEGINDATE=-7</b> .
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM-days	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM

Value	Description	Example
BOTM+days	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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
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.  To query volumes beginning with records changed to their current state a week ago, you can specify BEGINDATE=TODAY-7 or BEGINDATE=-7.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM-days	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+days	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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



Value	Description	Example
NOW+HH:MM or +HH:MM	The current time plus hours and minutes on the specified begin date	NOW+03:00 or +03:00.  If you issue <b>QUERY DRMEDIA</b> command at 9:00 with <b>BEGINTIME=NOW+03:00</b> or <b>BEGINTIME=+03:00</b> . The server displays volumes that were changed to their current state at 12:00 on the begin date that 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.  If you issue <b>QUERY DRMEDIA</b> command at 9:00 with <b>BEGINTIME=NOW-03:30</b> or <b>BEGINTIME=-03:30</b> . The server displays volumes that were changed to their current state at 5:30 on the begin date that 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
NOW+HH:MM or +HH:MM	The current time plus hours and minutes on the specified end date	NOW+03:00 or +03:00.  If you issue <b>QUERY DRMEDIA</b> command at 9:00 with <b>ENDTIME=NOW+03:00</b> or <b>ENDTIME=+03:00</b> , 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 -HH:MM	The current time minus hours and minutes on the specified end date	NOW-03:30 or -03:30  If you issue <b>QUERY DRMEDIA</b> command at 9:00 with <b>ENDTIME=NOW-03:00</b> or <b>ENDTIME=-03:00</b> , 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, the server selects the storage pools as follows:

- If the **SET DRMCOPYSTGPOOL** command was previously issued with valid copy storage pool names, the server 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, the server 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 storage pools that are specified with this parameter override those specified with the **SET DRMACTIVEDATASTGPOOL** command.

If this parameter is not specified, the server selects the storage pools as follows:

- If the **SET DRMACTIVEDATASTGPOOL** command was previously issued with valid active-data storage pool names, the server processes only those storage pools.
- If the **SET DRMACTIVEDATASTGPOOL** command has not been issued, or all of the active-data storage pools have been removed using the **SET DRMACTIVEDATASTGPOOL** command, the server processes all active-data storage 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 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, the server 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 is 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 Date/Time	Automated 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" 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 **Location** field is displayed when the volume is not mountable or when it's not in the library. The **Location** field is empty if the volume is mountable and is in the library.

**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 storage pool volume, the name of the active-data storage pool.

**Automated LibName**

The name of the automated library if the volume is in a library.

**Related commands**

*Table 224. Commands related to QUERY DRMEDIA*

<b>Command</b>	<b>Description</b>
BACKUP DB	Backs up the 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 onsite and offsite.
QUERY DRMSTATUS	Displays DRM system parameters.
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 DRMDBBACKUPEXPIREDDAYS** 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 DRMRPFEXPIREDDAYS** 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 225. Commands related to QUERY DRMSTATUS*

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
PREPARE	Creates a recovery plan file.
QUERY DRMEDIA	Displays information about disaster recovery volumes.
SET DRMCHECKLABEL	Specifies whether 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.

Table 225. Commands related to QUERY DRMSTATUS (continued)

Command	Description
SET DRMDBBACKUPEXPIREDDAYS	Specifies criteria for database backup series expiration.
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 DRMRPFEXPIREDDAYS	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.



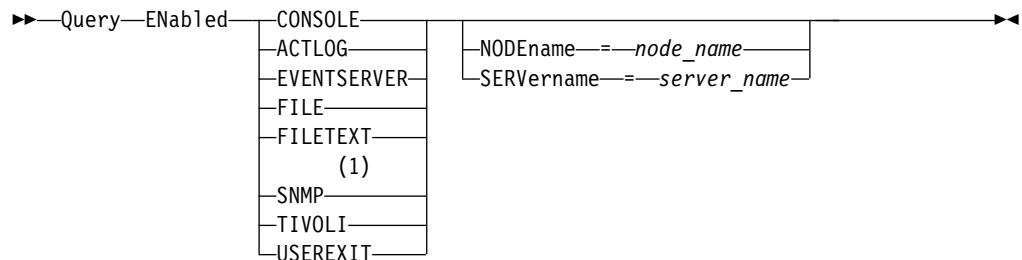
## QUERY ENABLED (Query enabled events)

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



### Notes:

- 1 This parameter is only available for the Linux operating system.

### 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 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 226. 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.

- “QUERY EVENT (Display administrative event schedules)” on page 757
- “QUERY EVENT (Display client schedules)” on page 750

*Table 227. Commands related to QUERY EVENT*

<b>Command</b>	<b>Description</b>
DEFINE SCHEDULE	Defines a schedule for a client operation or an administrative command.
DELETE EVENT	Deletes event records before 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.

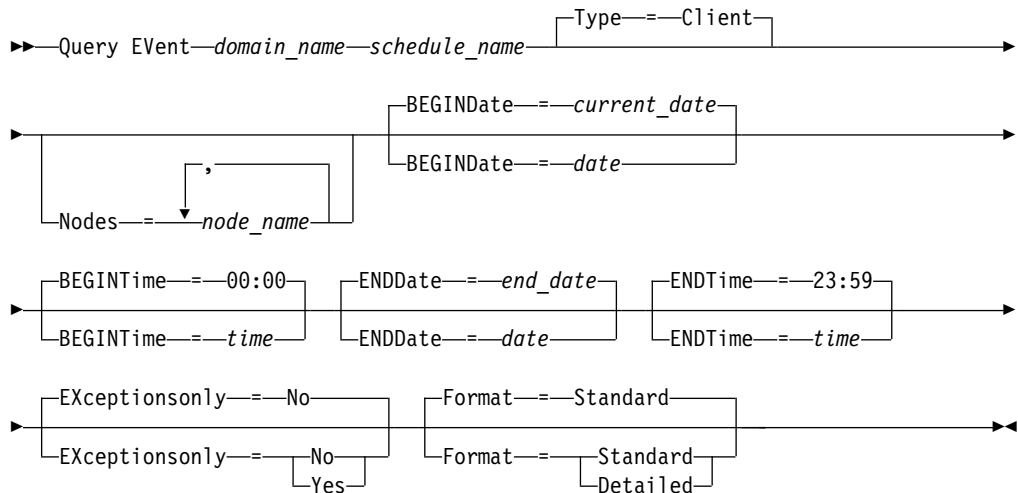
## QUERY EVENT (Display client schedules)

Use the **QUERY EVENT** command to display scheduled and completed events for selected clients.

### 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

Value	Description	Example
<b>TODAY</b>	The current date	TODAY
<b>TODAY+days or +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 or -days</b>	The current date minus days specified	TODAY-7 or -7.  To query events scheduled to start during the past seven days, specify BEGINDATE=TODAY-7 ENDDATE=TODAY or BEGINDATE=-7 ENDDATE=TODAY.
<b>EOLM (End Of Last Month)</b>	The last day of the previous month.	EOLM
<b>EOLM-days</b>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
<b>BOTM (Beginning Of This Month)</b>	The first day of the current month.	BOTM
<b>BOTM+days</b>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### **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
<b>HH:MM:SS</b>	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 or +HH:MM</b>	The current time plus hours and minutes on the specified begin date	NOW+03:00 or +03:00.  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.

Value	Description	Example
<b>NOW-HH:MM</b> or <b>-HH:MM</b>	The current time minus hours and minutes on the specified begin date	NOW-04:00 or -04:00.  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 ENDTIME=NOW or 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> or <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 <i>-days</i>	The current date minus days specified	TODAY-8 or -8.  To query events scheduled to start during a one-week period that ended yesterday, you can specify either BEGINDATE=TODAY-8 ENDDATE=TODAY-1 or BEGINDATE=-8 ENDDATE=-1.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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> or <b>+HH:MM</b>	The current time plus hours and minutes on the specified end date	NOW+03:00 or +03:00.  If you issue this command at 9:00 to query events scheduled to start 3 hours from now, you can specify either BEGINTIME=NOW ENDTIME=NOW+03:00 or BEGINTIME=NOW ENDTIME=+03:00.
<b>NOW-HH:MM</b> 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. You can specify one of the following values:

**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. The following values are possible:

#### Standard

Specifies that partial information for events displays.

#### Detailed

Specifies that complete information for events displays.

### Display partial information for unsuccessful events

Display partial information for all events that are scheduled for DOMAIN1 that did not run successfully. Limit the search to the client named JOE. Limit the events that are displayed to events 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 exceptiononly=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 755 for field descriptions.

## Display partial information for scheduled events for a client

Display complete information for all events that are scheduled for processing. Use the start time as 10 days previous to today, and the finish includes today.

```
query event * * begindate=today-10 enddate=today
```

Scheduled Start	Actual Start	Schedule Name	Node Name	Status
02/04/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/04/2013 14:00:00	02/04/2013 14:12:49	VDATAMVR1-IN1	VDATAMVR1-T1	Completed
02/04/2013 14:30:00	02/04/2013 14:33:10	VDATAMVR1-IN2	VDATAMVR1-T2	Completed
02/04/2013 15:00:00	02/04/2013 15:01:49	VDATAMVR1-IN3	VDATAMVR1-T3	Completed
02/04/2013 15:30:00	02/04/2013 15:42:00	VDATAMVR1-IN4	VDATAMVR1-T4	Completed
02/05/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/05/2013 14:00:00	02/05/2013 14:05:22	VDATAMVR1-F1	VDATAMVR1-F1	Completed
02/05/2013 14:30:00	02/05/2013 14:32:53	VDATAMVR1-F2	VDATAMVR1-F2	Failed 12
02/05/2013 15:00:00	02/05/2013 15:00:38	VDATAMVR1-F3	VDATAMVR1-F3	Completed
02/05/2013 15:30:00	02/05/2013 15:36:41	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/06/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/06/2013 14:00:00	02/06/2013 14:06:42	VDATAMVR1-F1	VDATAMVR1-F1	Completed
02/06/2013 14:30:00	02/06/2013 14:35:41	VDATAMVR1-F2	VDATAMVR1-F2	Completed
02/06/2013 15:00:00	02/06/2013 15:08:56	VDATAMVR1-F3	VDATAMVR1-F3	Completed
02/06/2013 15:30:00	02/06/2013 15:40:49	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/07/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/07/2013 14:00:00	02/07/2013 14:03:43	VDATAMVR1-F1	VDATAMVR1-F1	Completed
02/07/2013 14:30:00	02/07/2013 14:35:10	VDATAMVR1-F2	VDATAMVR1-F2	Completed
02/07/2013 15:00:00	02/07/2013 15:09:12	VDATAMVR1-F3	VDATAMVR1-F3	Completed
02/07/2013 15:30:00	02/07/2013 15:40:21	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/08/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/08/2013 14:00:00	02/08/2013 14:10:17	VDATAMVR1-F1	VDATAMVR1-F1	Completed
02/08/2013 14:30:00	02/08/2013 14:39:16	VDATAMVR1-F2	VDATAMVR1-F2	Completed
02/08/2013 15:00:00	02/08/2013 15:08:17	VDATAMVR1-F3	VDATAMVR1-F3	Completed
02/08/2013 15:30:00	02/08/2013 15:41:16	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/09/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/09/2013 14:02:16		VDATAMVR1-F1	VDATAMVR1-F1	Failed 12
02/09/2013 14:30:00	02/09/2013 14:44:26	VDATAMVR1-F2	VDATAMVR1-F2	Failed 12
02/09/2013 15:00:00	02/09/2013 15:06:24	VDATAMVR1-F3	VDATAMVR1-F3	Failed 12
02/09/2013 15:30:00	02/09/2013 15:32:18	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/11/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/11/2013 14:00:00	02/11/2013 14:01:05	VDATAMVR1-F1	VDATAMVR1-F1	Failed 12
02/11/2013 14:30:00	02/11/2013 14:31:42	VDATAMVR1-F2	VDATAMVR1-F2	Failed 12
02/11/2013 15:00:00	02/11/2013 15:06:17	VDATAMVR1-F3	VDATAMVR1-F3	Failed 12
02/11/2013 15:30:00	02/11/2013 15:30:19	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/12/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/12/2013 14:00:00	02/12/2013 14:03:37	VDATAMVR1-F1	VDATAMVR1-F1	Completed
02/12/2013 14:30:00	02/12/2013 14:33:07	VDATAMVR1-F2	VDATAMVR1-F2	Completed
02/12/2013 15:00:00	02/12/2013 15:03:56	VDATAMVR1-F3	VDATAMVR1-F3	Completed
02/12/2013 15:30:00	02/12/2013 15:36:44	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/13/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Missed
02/13/2013 14:00:00	02/13/2013 14:06:24	VDATAMVR1-F1	VDATAMVR1-F1	Completed
02/13/2013 14:30:00	02/13/2013 14:34:50	VDATAMVR1-F2	VDATAMVR1-F2	Completed
02/13/2013 15:00:00	02/13/2013 15:15:01	VDATAMVR1-F3	VDATAMVR1-F3	Completed
02/13/2013 15:30:00	02/13/2013 15:30:18	VDATAMVR1-F4	VDATAMVR1-F4	Completed
02/14/2013 14:00:00		SCHD_INCR-DM1	TSM_CET_DM1	Future
02/14/2013 14:00:00		VDATAMVR1-F1	VDATAMVR1-F1	Future
02/14/2013 14:30:00		VDATAMVR1-F2	VDATAMVR1-F2	Future
02/14/2013 15:00:00		VDATAMVR1-F3	VDATAMVR1-F3	Future

See "Field descriptions" on page 755 for field descriptions.

## Display detailed information for scheduled events for a client

Display the detailed information for events that are scheduled for processing by client DOC between the hours of 10:00 AM and 11:00 AM 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
begintime=10:00 endtime=11:00 enddate=11/01/2005
exceptionsonly=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 that is 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. The following values are possible:

#### Completed

Specifies that the scheduled event is completed.

**Failed** Specifies that the client reports a failure when you run the scheduled operation and successive retries failed.

#### 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 `dsmsched.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 passed and the schedule did not begin.

**Pending**

Specifies that the **QUERY EVENT** command was issued during the startup window for the event, but processing the scheduled operation did not begin.

**Restarted**

Specifies that the client has tried to process the scheduled operation again.

**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 is not found if the record was deleted or if the server was unavailable during the scheduled startup window (the schedule was never started). Records with Uncertain status are not stored 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, and is not 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 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).

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.

**Reason**

Specifies the reason for the return code.

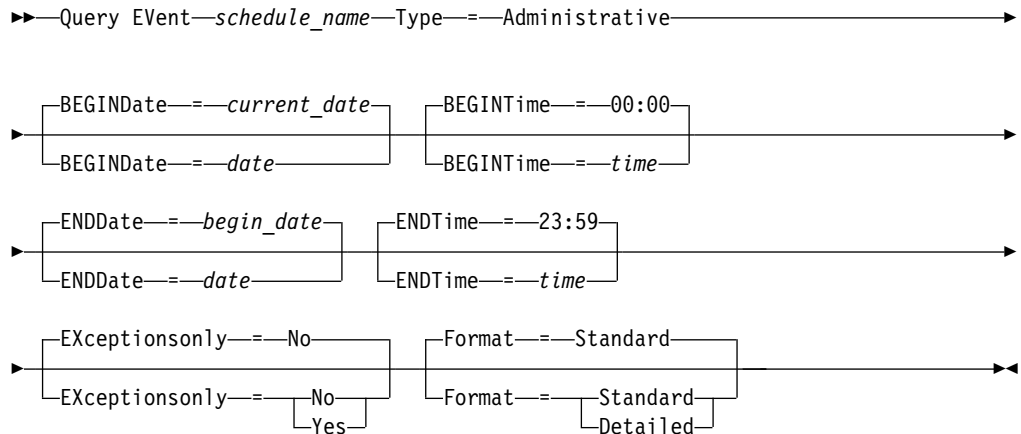
## QUERY EVENT (Display administrative event schedules)

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 <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 <i>-days</i>	The current date minus days specified	TODAY-7 or -7.  To query events scheduled to start during the past seven days, specify BEGINDATE=TODAY-7 ENDDATE=TODAY or BEGINDATE=-7 ENDDATE=TODAY.

Value	Description	Example
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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
NOW	The current time on the specified begin date	NOW
NOW+ <i>HH:MM</i> or + <i>HH:MM</i>	The current time plus hours and minutes on the specified begin date	NOW+03:00 or +03:00.  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.
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes on the specified begin date	NOW-04:00 or -04:00.  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 ENDTIME=NOW or 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> or <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 <i>-days</i>	The current date minus days specified	TODAY-8 or -8.  To query events scheduled to start during a one-week period that ended yesterday, you can specify either BEGINDATE=TODAY-8 ENDDATE=TODAY-1 or BEGINDATE=-8 ENDDATE=-1.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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> or <i>+HH:MM</i>	The current time plus hours and minutes on the specified end date	NOW+03:00 or +03:00.  If you issue this command at 9:00 to query events scheduled to start 3 hours from now, you can specify either BEGINTIME=NOW ENDTIME=NOW+03:00 or BEGINTIME=NOW ENDTIME=+03:00.
<b>NOW-HH:MM</b> or <i>-HH:MM</i>	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. You can specify one of the following values:

**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
```

Scheduled Start	Actual Start	Schedule Name	Status
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.

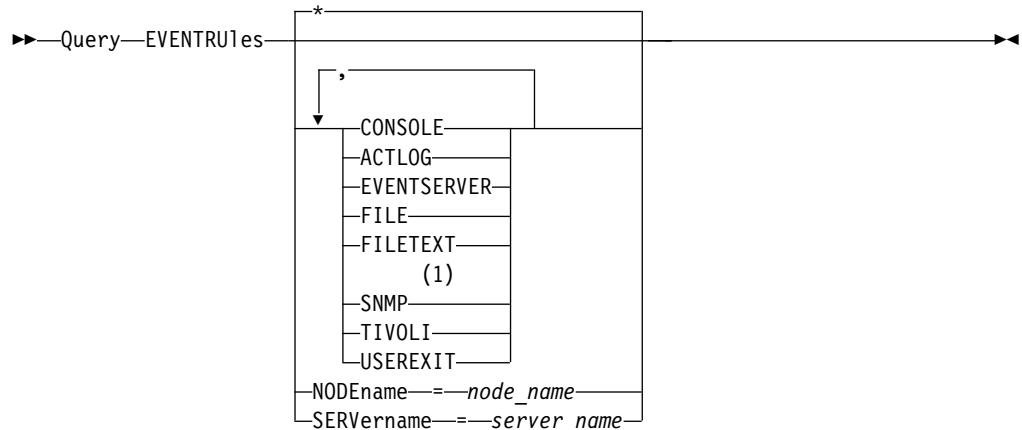
## QUERY EVENTRULES (Query rules for server or client events)

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



### Notes:

- 1 This parameter is only available for the Linux operating system.

### 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.

### 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

## Field descriptions

### Date/Time

Specifies the date and time when the event was enabled or disabled.

### Client Event Rules

Specifies client events that were enabled or disabled for the specified receivers.

### Server Event Rules

Specifies server events that were enabled or disabled for the specified receivers.



## Related commands

*Table 228. Commands related to QUERY ENABLED*

<b>Command</b>	<b>Description</b>
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 229. 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)

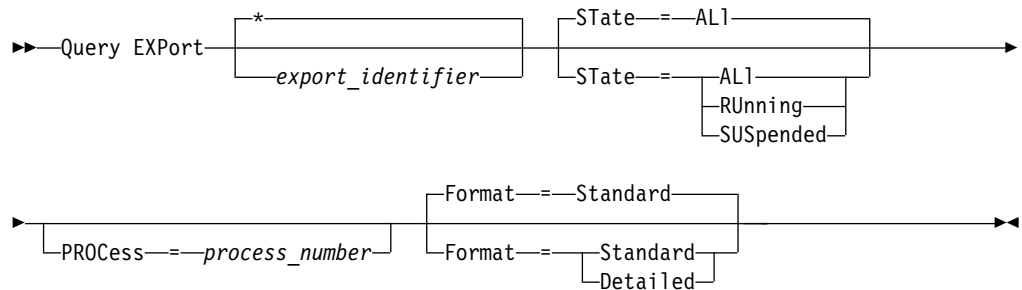
Use this command to list all restartable export operations. A restartable export is a server-to-server export operation whose FILEDATA value is not NONE. Only active server-to-server export operations that can be suspended are displayed.

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.

##### **RUning**

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 filesystem=c\$ nametype=unicode filedata=all durunits= indefinite toserver=athens exportid=MYEXPORTNODE
EXPORT_HOME_DIRS	01/25/2007 09:30:03	Running	11	Export NODE n2,n3,n4 filesystem=/home nametype= server filedata=all durunits= indefinite toserver=athens exportid=EXPORT_HOME_DIRS
EXPORT_NODE_0001	01/25/2007 14:30:33	Running Not Suspendible	--	Export NODE n5,n6,n7 filesystem=d\$ nametype=unicode filedata=archive durunits= indefinite toserver=athens

See “Field descriptions” on page 768 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 filesystem=c\$ nametype=unicode filedata=all toserver=athens exportid=MYEXPORTNODE

See “Field descriptions” on page 768 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 768 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 of bytes transmitted so far to the target server for this export operation.

**Total files to be transferred**

The total number of files transmitted so far to the target server for this export operation.

**Files remaining**

The total number of files remaining to be transmitted to the target server for this export operation.



## Related commands

Table 230. 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 or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
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)

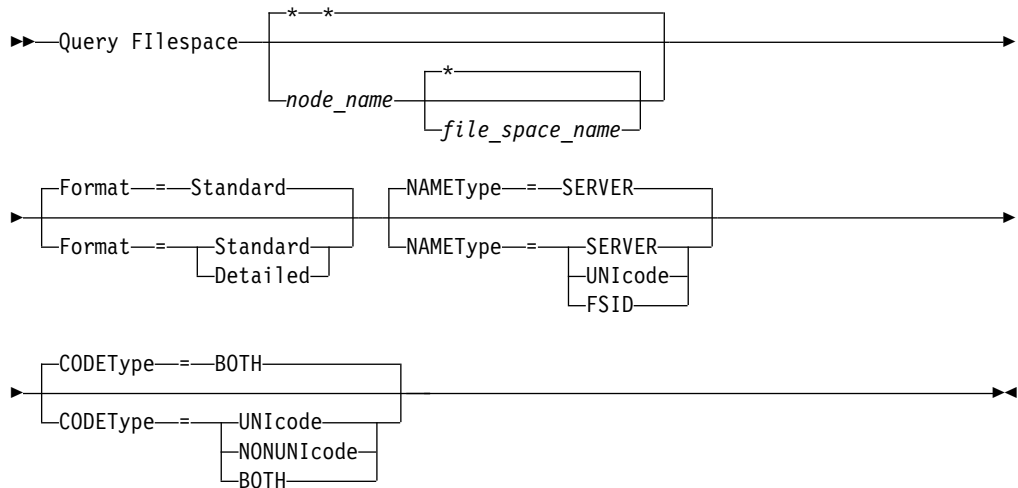
Use this command to display information about file spaces that belong to a client node. The output from this command includes the results of the last incremental backup or replication.

**Tip:** If a node has more than one file space, you can issue a **DELETE FILESPACE** command for one of the file spaces. However, if you issue a **QUERY FILESPACE** command for the node during the deletion process, the output shows no file spaces. To obtain accurate information about remaining file spaces, issue the **QUERY FILESPACE** command after the deletion process ends.

### 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.

If a server includes clients that use Unicode-enabled file spaces, the server might have to convert the name that you enter. For example, the server might have to convert the file space name that you enter from the server code page to Unicode. For more information, see the **NAMETYPE** parameter. If you do not specify a file space name, or if you 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.

**Format**

Specifies how the information is displayed. This parameter is optional. The default value is **STANDARD**. You can specify one of the following values:

**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 that have Windows, Macintosh OS X, and NetWare operating systems.

Use this parameter only when you enter a partly qualified or fully qualified file space name. The default value is **SERVER**. You can specify one of the following values:

**SERVER**

The server uses the server code page to interpret the file space names.

**UNICODE**

The server converts the file space name that is 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 code page. Conversion can fail if the string includes characters that are not available in the server code page, or if the server has problems 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. You can specify one of the following values:

**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 that are associated with all client nodes.

```
query filespace
```

Node Name	Filespace Name	FSID	Platform	Filespace Type	Is Filespace Unicode?	Capacity	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 775 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	Pct Util
NAS1	/HomeDir	1	NetApp	WAFL (VFS)	No	2,502.3	75.2

See “Field descriptions” on page 775 for field descriptions.

**Important:** You might not see the expected results after you request a detailed format because several fields must be completed by the API application. These fields include:

- File space type
- Platform
- Capacity
- Pct Util
- Last backup start Date/Time
- Last backup completion Date/Time

For more information about specific fields that are updated by the API, see the *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 about 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
Collocation Group Name: FSGRP1
Platform: WinNT
Filespace Type: NTFS
Is Filespace Unicode?: Yes
Capacity: 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:
Last Replication Start Date/Time: 12/02/2012, 12:42:00
Days Since Last Node Replication Started: 30
Last Replication Completion Date/Time: 12/02/2012, 12:42:00
Days Since Last Replication Completed: 30
Last Backup Date/Time From Client (UTC): 06/02/2013, 09:10:00
Last Archive Date/Time From Client (UTC): 06/02/2013, 09:10:00
Backup Replication Rule Name: ACTIVE_DATA
Backup Replication Rule State: ENABLED
Archive Replication Rule Name: DEFAULT
Archive Replication Rule State: ENABLED
Space Management Replication Rule Name: NONE
Space Management Replication Rule State: DISABLED
At-risk type: Custom interval
At-risk interval: 2,222
Decommissioned: No
Decommissioned Date:
MAC Address:

```

See “Field descriptions” for field descriptions.

## Field descriptions

**Important:** You might not see the expected results after requesting a detailed format because several fields must be completed by the API application. These fields include:

- Filespace Type
- Platform
- Capacity
- Pct Util
- Last Backup Start Date/Time
- Last Backup Completion Date/Time

For more information about specific fields that are updated by the API, see the *IBM Tivoli Storage Manager: Using the Application Programming Interface*.

### Node Name

Specifies the name of the client node.

### Filespace Name

The name of the file space that belongs to the node.

File space names can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

**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.

**Collocation Group Name**

The name of the collocation group, if any, to which the file space belongs.

**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**

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 field represents 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 used 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.

**Last Replication Start Date/Time**

Specifies the date and time that the last replication of file space data started.

**Days Since Last Replication Started**

Specifies the number of days since the last replication of file space data started.

**Last Replication Completion Date/Time**

Specifies the date and time that the last replication of file space data ended.

**Days Since Last Replication Completed**

Specifies the number of days since the last replication of file space data ended.

**Last Backup Date/Time From Client (UTC)**

The date and time, in Universal Time Coordinates (UTC), of the last backup operation for this file space.

**Last Archive Date/Time From Client (UTC)**

The date and time, in Universal Time Coordinates (UTC), of the last archive operation for this file space.

**Backup Replication Rule Name**

Specifies the replication rule that applies to backup data in the file space. The following values are possible:

**ALL\_DATA**

Replicates active and inactive backup data. The data is replicated with a normal priority.

**ACTIVE\_DATA**

Replicates only active backup data. The data is replicated with a normal priority.

**Attention:** If you specify **ACTIVE\_DATA** and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the **FORCERECONCILE=YES** parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

**ALL\_DATA\_HIGH\_PRIORITY**

Replicates active and inactive backup data. The data is replicated with a high priority.

**ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the **ACTIVE\_DATA** replication rule except data is replicated with a high priority.

**DEFAULT**

Replicates backup data according to the client node rule for backup

data. If the client node rule for backup data is DEFAULT, backup data is replicated according to the server rule for backup data.

**NONE**

Backup data in the file space is not replicated.

**Backup Replication Rule State**

Specifies whether replication of backup data in the file space is enabled or disabled. If the state is ENABLED, backup files are eligible for replication. If the state is DISABLED, backup files are not eligible for replication.

**Archive Replication Rule Name**

Specifies the replication rule that applies to archive data in the file space. The following values are possible:

**ALL\_DATA**

Replicates archive data. The data is replicated with a normal priority.

**ALL\_DATA\_HIGH\_PRIORITY**

Replicates archive data. The data is replicated with a high priority.

**DEFAULT**

Replicates archive data according to the client rule for archive data. If the client rule for archive data is DEFAULT, archive data is replicated according to the server rule for archive data.

**NONE**

Archive data in the file space is not replicated.

**Archive Replication Rule State**

Specifies whether replication of archive data in the file space is enabled or disabled. If the state is ENABLED, archive files are eligible for replication. If the state is DISABLED, archive files are not eligible for replication.

**Space Management Replication Rule Name**

Specifies the replication rule that applies to space-managed data in the file space. The following values are possible:

**ALL\_DATA**

Replicates space-managed data. The data is replicated with a normal priority.

**ALL\_DATA\_HIGH\_PRIORITY**

Replicates space-managed data. The data is replicated with a high priority.

**DEFAULT**

Replicates space-managed data according to the client rule for space-managed data. If the client rule for space-managed data is DEFAULT, space-managed data is replicated according to the server rule for space-managed data.

**NONE**

Space-managed data in the file space is not replicated.

**Space Management Replication Rule State**

Specifies whether replication of space-managed data in the file space is enabled or disabled. If the state is ENABLED, space-managed files are eligible for replication. If the state is DISABLED, space-managed files are not eligible for replication.



**At-risk type**

Specifies the at-risk evaluation type. Values can be Default, Bypassed, or Custom. Default indicates that the node is evaluated with the same interval that was specified for the nodes classification by the **SET STATUSATRISKINTERVAL** command. Bypassed indicates that the node is not evaluated for at-risk status by the status monitor. Custom indicates that the node is evaluated with the interval that was specified by the **SET VMATRISKINTERVAL** command, rather than the interval that was specified by the **SET STATUSATRISKINTERVAL** command.

**At-risk interval**

Specifies the amount of time, in hours, between client backup activity before the status monitor considers the client at-risk. This field applies only when the at-risk type is Custom.

**Decommissioned**

Specifies whether the virtual machine that the file space represents is decommissioned.

**Decommissioned Date**

Specifies the date that the virtual machine that the file space represents was decommissioned.

**MAC Address**

Specifies the media access control (MAC) address of the file spaces backed up for VMWare virtual machines. In the case where the virtual machine has multiple MAC addresses this is the lowest valued address.

**Related commands**

*Table 231. Commands related to QUERY FILESPACE*

Command	Description
DEFINE VIRTUALFSMAPPING	Define a virtual file space mapping.
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
REGISTER NODE	Defines a client node 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 file space on the server.
UPDATE FILESPACE	Changes file-space node-replication rules.
UPDATE NODE	Changes the attributes that are associated with a client node.

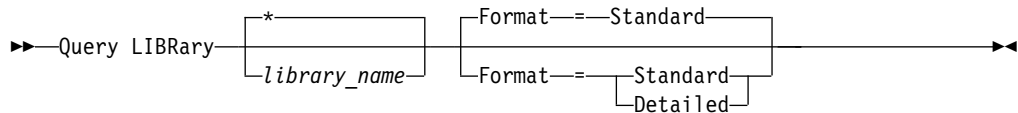
## QUERY LIBRARY (Query a library)

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 781 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 that is displayed in this field applies only to an IBM 3494 or 3495 Tape Library Dataserver.

### Scratch Category

The category number to use for scratch volumes in the library.

The information that is displayed in this field applies only to an IBM 3494 or 3495 Tape Library Dataserver.

### WORM Scratch Category

The category number that is used for WORM scratch volumes in the library.

The information that is displayed in this field applies only 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 that is set for mount retention in the device class that is associated with this external 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 that is being queried.

**AutoLabel**

Specifies whether the server attempts to automatically label tape volumes.

**Relabel Scratch**

Specifies whether the server relabels volumes that were deleted and returned to scratch.

**Last Update by (administrator)**

Who completed the last update to the library.

**Last Update Date/Time**

The date and time when the last update occurred.

**Related commands**

*Table 232. Commands related to QUERY LIBRARY*

<b>Command</b>	<b>Description</b>
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.

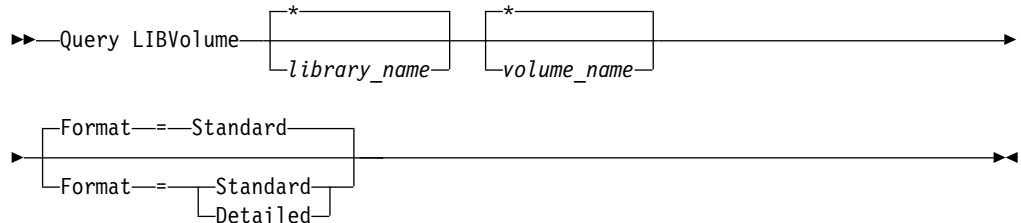
## QUERY LIBVOLUME (Query a library volume)

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 784 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 784 for field descriptions.

```
query libvolume jjy008 format=detailed
```

```

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 233. 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 233. Commands related to **QUERY LIBVOLUME** (continued)

<b>Command</b>	<b>Description</b>
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.

To display the license information, issue the following command:

```
query license
```

The following example output is displayed:

```
ANR2017I Administrator SERVER_CONSOLE issued command: QUERY LICENSE

                Last License Audit: 02/14/2013 14:28:08
                Number of TDP for Oracle in use: 0
                Number of TDP for Oracle in try buy mode: 0
                Number of TDP for MS SQL Server in use: 0
Number of TDP for MS SQL Server in try buy mode: 0
                Number of TDP for MS Exchange in use: 0
                Number of TDP for MS Exchange in try buy mode: 0
                Number of TDP for Lotus Notes in use: 12
                Number of TDP for Lotus Notes in try buy mode: 0
                Number of TDP for Lotus Domino in use: 0
                Number of TDP for Lotus Domino in try buy mode: 0
                Number of TDP for Informix in use: 1
                Number of TDP for Informix in try buy mode: 0
                Number of TDP for SAP R/3 in use: 0
                Number of TDP for SAP R/3 in try buy mode: 0
                Number of TDP for ESS in use: 0
                Number of TDP for ESS in try buy mode: 0
                Number of TDP for ESS R/3 in use: 0
                Number of TDP for ESS R/3 in try buy mode: 0
                Number of TDP for EMC Symmetrix in use: 0
                Number of TDP for EMC Symmetrix in try buy mode: 0
                Number of TDP for EMC Symmetrix R/3 in use: 6
                Number of TDP for EMC Symmetrix R/3 in try buy mode: 0
                Number of TDP for WAS in use: 0
                Number of TDP for WAS in try buy mode: 0
                Is IBM System Storage Archive Manager in use?: No
                Is IBM System Storage Archive Manager licensed?: Yes
                Is Tivoli Storage Manager Basic Edition in use: Yes
                Is Tivoli Storage Manager Basic Edition licensed: Yes
                Is Tivoli Storage Manager Extended Edition in use: No
                Is Tivoli Storage Manager Extended Edition licensed: Yes
                Server License Compliance: Valid
```

### Field descriptions

#### Last License Audit

Specifies the date and time when the last license audit occurred.



**Number of TDP for Oracle in use**

Specifies the number of TDP for Oracle that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for Oracle in try buy mode**

Specifies the number of TDP for Oracle that are in try buy mode.

**Number of TDP for MS SQL Server in use**

Specifies the number of TDP for MS SQL Server that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for MS SQL Server in try buy mode**

Specifies the number of TDP for MS SQL Server that are in try buy mode.

**Number of TDP for MS Exchange in use**

Specifies the number of TDP for MS Exchange that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for MS Exchange in try buy mode**

Specifies the number of TDP for MS Exchange that are in try buy mode.

**Number of TDP for Lotus Notes® in use**

Specifies the number of TDP for Lotus Notes that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for Lotus Notes in try buy mode**

Specifies the number of TDP for Lotus Notes that are in try buy mode.

**Number of TDP for Lotus Domino in use**

Specifies the number of TDP for Lotus Domino that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for Lotus Domino in try buy mode**

Specifies the number of TDP for Lotus Domino that are in try buy mode.

**Number of TDP for Informix in use**

Specifies the number of TDP for Informix that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for Informix in try buy mode**

Specifies the number of TDP for Informix that are in try buy mode.

**Number of TDP for SAP R/3 in use**

Specifies the number of TDP for SAP R/3 that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for SAP R/3 in try buy mode**

Specifies the number of TDP for SAP R/3 that are in try buy mode.

**Number of TDP for ESS in use**

Specifies the number of TDP for ESS that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for ESS in try buy mode**

Specifies the number of TDP for ESS that are in try buy mode.

**Number of TDP for ESS R/3 in use**

Specifies the number of TDP for ESS R/3 that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for ESS R/3 in try buy mode**

Specifies the number of TDP for ESS R/3 that are in try buy mode.

**Number of TDP for EMC Symmetrix in use**

Specifies the number of TDP for EMC Symmetrix that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for EMC Symmetrix in try buy mode**

Specifies the number of TDP for EMC Symmetrix that are in try buy mode.

**Number of TDP for EMC Symmetrix R/3 in use**

Specifies the number of TDP for EMC Symmetrix R/3 that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for EMC Symmetrix R/3 in try buy mode**

Specifies the number of TDP for EMC Symmetrix R/3 that are in try buy mode.

**Number of TDP for WAS in use**

Specifies the number of TDP for WAS that are in use. A product is in use if you purchased the product and registered the license.

**Number of TDP for WAS in try buy mode**

Specifies the number of TDP for WAS that are in try buy mode.

**Is IBM System Storage Archive Manager in use ?**

Specifies whether the IBM System Storage Archive Manager is in use. A product is in use if you purchased the product and registered the license.

**Is IBM System Storage Archive Manager licensed ?**

Specifies whether the IBM System Storage Archive Manager is licensed.

**Is IBM Tivoli Storage Manager Basic Edition in use**

Specifies whether the IBM Tivoli Storage Manager Basic Edition is in use. A product is in use if you purchased the product and registered the license.

**Is IBM Tivoli Storage Manager Basic Edition licensed**

Specifies whether the IBM Tivoli Storage Manager Basic Edition is licensed.

**Is IBM Tivoli Storage Manager Extended Edition in use**

Specifies whether the Tivoli Storage Manager Extended Edition is in use. A product is in use if you purchased the product and registered the license.

**Is IBM Tivoli Storage Manager Extended Edition licensed**

Specifies whether the Tivoli Storage Manager Extended Edition is licensed.

**Server License Compliance**

Specifies whether the server license is valid.

**Related commands**

*Table 234. Commands related to QUERY LICENSE*

Command	Description
AUDIT LICENSES	Verifies compliance with defined licenses.
QUERY AUDITOCCUPANCY	Displays the server storage utilization for a client node.
QUERY NODE	Displays partial or complete information about one or more clients.
QUERY PVUESTIMATE	Displays processor value unit estimates. <b>Remember:</b> The <b>QUERY PVUESTIMATE</b> command reports licenses by providing PVU information on a per-node basis for server devices.

Table 234. Commands related to **QUERY LICENSE** (continued)

<b>Command</b>	<b>Description</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REGISTER LICENSE	Registers a license with the Tivoli Storage Manager server.
REGISTER NODE	Defines a client node to the server and sets options for that user.
SET CPUINFOREFRESH	Specifies the number of days between client scans for workstation information used for PVU estimates.
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.
UPDATE NODE	Changes the attributes that are associated with a client node.

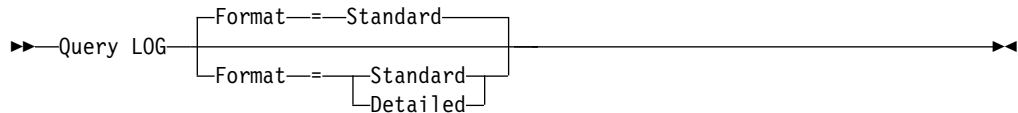
## QUERY LOG (Display information about the recovery log)

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. The following values are possible:

#### 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” on page 791 for field descriptions.

```
query log
```

Total Space (MB)	Used Space (MB)	Free Space (MB)
38,912	543.3	38,368.7

### Example: Display detailed information about the recovery log

Display detailed information about the recovery log. See “Field descriptions” on page 791 for field descriptions.

```
query log format=detailed
```

```

Active Log Directory : /actlog
Total Space (MB): 524,032
Used Space (MB): 3,517
Free Space (MB): 520,515

Total Size of File System (MB): 564,443
Used Space on File System (MB): 527,049
Free Space on File System (MB): 8,722

Archive Log Directory : /archlog
Total Size of File System (MB): 603,751.82
Used Space on File System (MB): 80,642.30
Free Space on File System (MB): 523,109.52
Archive Log Compressed : Yes

Mirror Log Directory : /mirrorlog
Total Size of File System (MB): 564,443
Used Space on File System (MB): 527,049
Free Space on File System (MB): 8,722

Archive Failover Log Directory : /archfaillog
Total Size of File System (MB): 301,372.06
Used Space on File System (MB): 44,741.80
Free Space on File System (MB): 256,630.26

```

## 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.

### Free Space

Specifies the amount of active log space that is not being used by uncommitted transactions, in megabytes.

### Total Size of File System

Specifies the total size of the file system, in megabytes.

### Space Used on File System

Specifies the amount of used space on the file system, in megabytes.

### Free Space on File System

Specifies the amount of space that is available on the file system, in megabytes.

### Archive Log Compressed

Specifies whether the archive logs are compressed.

### 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.

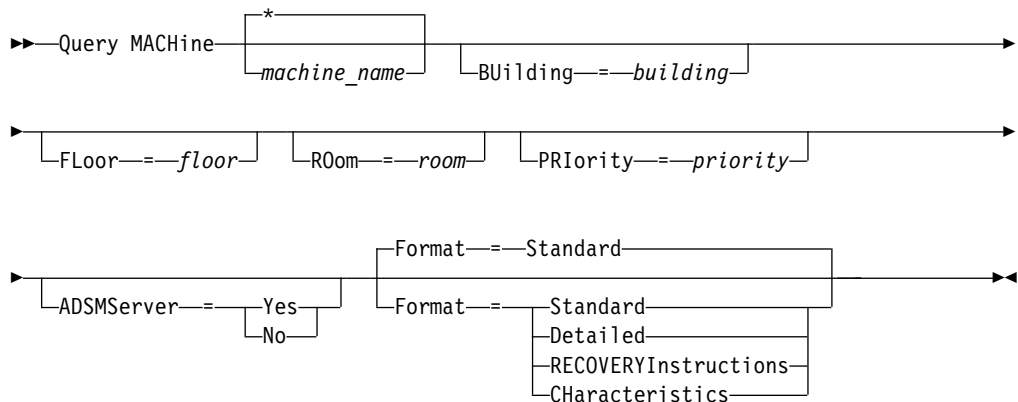
Tivoli Storage Manager displays information for multiple machines in the following order:

- 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.

#### **R0om**

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 794 for field descriptions.

```
query machine MACH1
```

Machine Name	Machine Priority	Building	Floor	Room	Node Name	Recovery 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 794 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 235. Commands related to QUERY MACHINE*

Command	Description
DEFINE MACHINE	Defines a machine for DRM.
DEFINE MACHNODEASSOCIATION	Associates a 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 Tivoli Storage Manager database.
UPDATE MACHINE	Changes the information for a machine.



## QUERY MEDIA (Query sequential-access storage pool media)

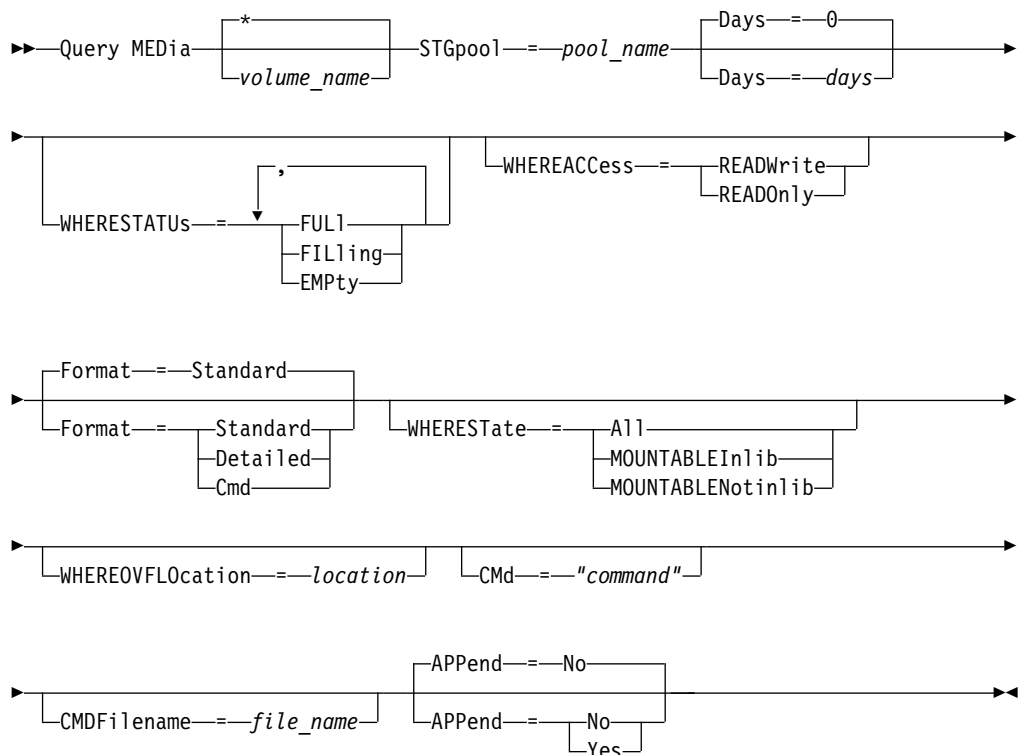
Use this command to display information about the sequential-access primary and copy storage pool volumes moved by the **MOVE MEDIA** command.

### 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.

The **QUERY MEDIA** command displays only volumes with an **ACCESS MODE** value of **READONLY** or **READWRITE**.

### 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 800 for field descriptions.

```
query media * stgpool=archive wherestatus=full, filling
```

Volume Name	State	Location	Automated LibName
TAPE01	Mountable in Library		LIB3494
TAPE03	Mountable not in Lib.	Room1234/Bldg31	
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 800 for field descriptions.

```
query media wherestate=mountablenotinlib stgpool=onsite*  
wherestatus=full format=detailed
```

```
Volume Name: TAPE21  
State: Mountable not in library  
Volume 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  
Volume 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.

### 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 236. Commands related to QUERY MEDIA*

Command	Description
MOVE MEDIA	Moves storage pool volumes that are managed by an automated library.

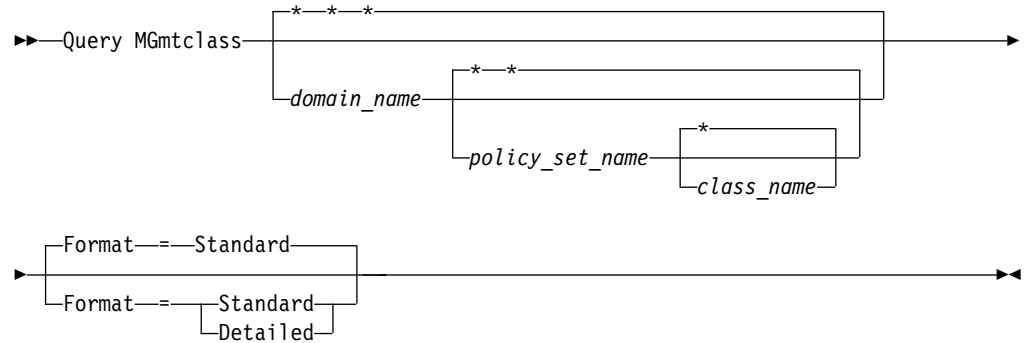
## QUERY MGMTCLASS (Query a management class)

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 802 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

To display information about management classes in a specific policy domain, for example the domain ENGPOLDOM, issue the following command:

```
query mgmtclass engpoldom * *
```

**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: SPACEMGPOOL
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.

**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 237. 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 MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)

Use this command to display information about alert monitoring and server status settings.

### Privilege class

Any administrator can issue this command.

### Syntax

►► Query MONITORSEttings ◀◀

### Display monitoring settings

Display details about the monitoring settings. See Field descriptions for more details.

query monitorsettings

Example output:

```
Monitor Status: On
Status Refresh Interval (Minutes): 5
Status Retention (Hours): 48
Monitor Message Alerts: On
Alert Update Interval (Minutes): 10
Alert to Email: On
Send Alert Summary to Administrators: On
Alert from Email Address: DJADMIN@MYDOMAIN.COM
Alert SMTP Host: DJHOST.MYDOMAIN.COM
Alert SMTP Port: 25
Alert Active Duration (Minutes): 480
Alert Inactive Duration (Minutes): 480
Alert Closed Duration (Minutes): 60
Monitoring Admin: ADMIN
Monitored Group: MONGROUP
Monitored Servers: SERVER2
At-Risk Interval for Applications: 24
Skipped files as At-Risk for Applications?: Yes
At-Risk Interval for Virtual Machines: 24
Skipped files as At-Risk for Virtual Machines?: Yes
At-Risk Interval for Systems: 24
Skipped files as At-Risk for Systems?: Yes
```

### Field descriptions

#### Monitor status

Specifies whether alert monitoring on the server is enabled or disabled.

#### Status Refresh Interval (Minutes)

Specifies the number of minutes between intervals that the monitoring server gathers event data.

#### Status Retention (Hours)

Specifies the number of hours that status monitoring indicators are retained.

#### Monitor Message Alerts

Specifies whether alerts are sent to administrators by email.

**Alert Update Interval (Minutes)**

Specifies the length of time, in minutes, that the alert monitor waits before the alert is updated and pruned on the server.

**Alert to Email**

Specifies whether alerts are sent to administrators by email.

**Send Alert Summary to Administrators**

Specifies the administrators that receive a summary of existing alerts on the server in an email.

**Alert from Email Address**

Specifies the email address of the sender.

**Alert SMTP Host**

Specifies the Simple Mail Transfer Protocol (SMTP) host mail server that is used to send alerts by email.

**Alert SMTP Port**

Specifies the SMTP mail server port that is used to send alerts by email.

**Alert Active Duration (Minutes)**

Specifies how long, in minutes, an alert remains active.

**Alert Inactive Duration (Minutes)**

Specifies how long, in minutes, an alert remains inactive.

**Alert Closed Duration (Minutes)**

Specifies how long, in minutes, an alert remains closed before it is deleted from the server.

**Monitoring Admin**

Specifies the name of the monitoring administrator that is used to connect to the servers in the monitored group.

**Monitored Group**

Specifies the name of the monitored server group.

**Monitored Servers**

Specifies the names of the servers in the monitored server group. The monitor settings might be different on each monitored server. If so, issue the query command for each server to display the monitoring settings.

**At-Risk Interval for Applications**

Specifies how long, in hours, an applications client can log no activity before it is considered at-risk.

**Skipped files as At-Risk for Applications?**

Specifies that the server considers skipped files, by the client as a failure, and marks the client at-risk.

**At-Risk Interval for Virtual Machines**

Specifies how long, in hours, a virtual client can log no activity before it is considered at-risk.

**Skipped files as At-Risk for Virtual Machines?**

Specifies that the server considers skipped files, by the client as a failure and marks the client at-risk.

**At-Risk Interval for Systems**

Specifies how long, in hours, a systems client can log no activity before it is considered at-risk.

## Skipped files as At-Risk for Systems?

Specifies that the server considers skipped files, by the client as a failure, and marks the client at-risk.

## Related commands

Table 238. Commands related to QUERY MONITORSETTINGS

Command	Description
"DEFINE ALERTTRIGGER (Define an alert trigger)" on page 124	Associates specified messages to an alert trigger.
"DELETE ALERTTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.
"DELETE GRPMEMBER (Delete a server from a server group)" on page 416	Deletes a server from a server group.
"DELETE SERVER (Delete a server definition)" on page 439	Deletes the definition of a server.
"QUERY ALERTSTATUS (Query the status of an alert)" on page 680	Displays information about alerts that have been issued on the server.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"SET ALERTMONITOR (Set the alert monitor to on or off)" on page 1091	Specifies whether alert monitoring is set to on or off.
"SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176	Specifies whether to enable client at-risk activity interval evaluation
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)" on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.
"UPDATE ALERTSTATUS (Update the status of an alert)" on page 1205	Updates the status of a reported alert.

## QUERY MONITORSTATUS (Query the monitoring status)

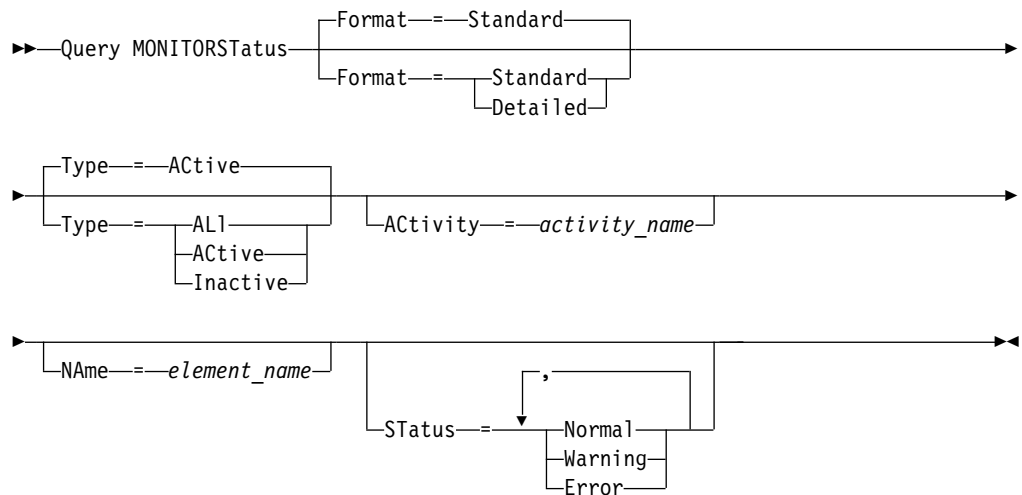
Use this command to display monitoring messages that are within the defined status retention period.

You can limit the output to a specified status, such as only messages with a status of active. If you do not specify any parameters, all messages are displayed.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### Format

Specifies the amount of information that is displayed. The default value is STANDARD. Specify one of the following values:

##### Standard

Specifies that only partial information is displayed for the specified messages.

##### Detailed

Specifies that all information is displayed for the specified messages.

#### Type

This parameter restricts the output to only messages with the specified type value. Specify one of the following values:

##### All

Displays all information.

##### Active

Displays all active messages. This is the default value.

##### Inactive

Displays all inactive messages.

**ACTivity**

Specifies the activity that you want to query. See the **DEFINE STATUSTHRESHOLD** command for details on available activities to query.

**NAME**

Specifies the name that you want to query. The NAME value refers to the name of the element with the specified activity. For example, a status indicator that contains information about a storage pool that is called backuppool has the NAME set to BACKUPPOOL.

**STATUS**

Specifies the status of the messages that you want to query. You can specify multiple status values in a list by separating the values with commas and no intervening spaces. If you do not specify a value for this parameter, information for all status values is displayed. Specify one of the following values:

**Normal**

Displays all messages with a normal status.

**Warning**

Displays all messages with a warning status.

**Error**

Displays all messages with an error status.

**Display monitoring settings**

Display details about the monitoring status.

Query MONITORStatus type=active

Example output:

```

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Name: CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: USED CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Name: USED CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: CAPACITY OF PRIMARY TAPE STORAGE
Element Name: CAPACITY OF PRIMARY TAPE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: USED CAPACITY OF PRIMARY TAPE STORAGE
Element Name: USED CAPACITY OF PRIMARY TAPE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL

```

## Display monitoring settings

Display details about the monitoring status.

```
query monitorstatus f=d type=active
```

Example output:

```
Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Name: CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL
Element Details:
Primary Repair Suggestion:
First Alternate Repair Suggestion:
Second Alternate Repair Suggestion:

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: USED CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Name: USED CAPACITY OF PRIMARY DISK AND FILE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL
Element Details:
Primary Repair Suggestion:
First Alternate Repair Suggestion:
Second Alternate Repair Suggestion:

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: CAPACITY OF PRIMARY TAPE STORAGE
Element Name: CAPACITY OF PRIMARY TAPE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL
Element Details:
Primary Repair Suggestion:
First Alternate Repair Suggestion:
Second Alternate Repair Suggestion:

Server Name: SERVER1
Activity Date: 03/05/2013 15:57:37
Activity Name: USED CAPACITY OF PRIMARY TAPE STORAGE
Element Name: USED CAPACITY OF PRIMARY TAPE STORAGE
Element Numeric Value: 0
Element String Value:
Element State: NORMAL
Element Details:
Primary Repair Suggestion:
First Alternate Repair Suggestion:
Second Alternate Repair Suggestion:
```

## Related commands

Table 239. Commands related to **QUERY MONITORSTATUS**

Command	Description
“DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)” on page 326	Defines a status monitoring threshold.
“DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)” on page 442	Deletes a status monitoring threshold.
“QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804	Displays information about monitoring alerts and server status settings.

Table 239. Commands related to **QUERY MONITORSTATUS** (continued)

Command	Description
"QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932	Displays information about a status monitoring thresholds.
"SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176	Specifies whether to enable client at-risk activity interval evaluation
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)" on page 1180	Specifies the refresh interval for status monitoring.
"SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)" on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
"UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389	Changes the attributes of an existing status monitoring threshold.



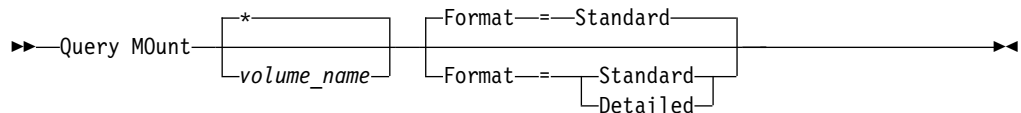
## QUERY MOUNT (Display information on mounted sequential access volumes)

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.

#### 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 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/rmt/1m), 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 is 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 is read/write (R/W), even if it is only being read.
2. In a storage pool that is associated with the `FILE` or `CENTERA` device type, the server can complete 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` or `CENTERA` can appear to be mounted more than once.

3. In the message ANR8448I, the drive name is listed as UNKNOWN for volumes of the FILE device type with a non-shared device class. The reason is that no drive is associated with the volumes; drive names are shown in the file-based library.
4. If you issue the **QUERY MOUNT** command while the drive is being cleaned, the command output continues to show a DISMOUNTING status for the dismounted volume until the cleaning completes.

### Example: Display detailed information about mounted sequential volumes

Display details about mounted volumes.

```
query mount format=detailed
```

```
ANR2017I Administrator SERVER_CONSOLE issued command: QUERY
MOUNT format=detailed
ANR8487I Mount point in device class FILE is waiting for the
volume mount to
complete -- owning server: SERVER1, status: WAITING FOR VOLUME
(session: 0, process: 1).
ANR8488I LTO volume 015005L4 is mounted R/W in drive IBMVTL1
(/dev/rmt37) -- owning
server: SERVER1, status: IN USE (session: 0, process: 2).
ANR8486I Mount point in device class FILE is reserved -- owning
server: SERVER1,
status: RESERVED (session: 5, process: 0).
ANR8334I      3 matches found.
```

### Related commands

Table 240. 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)

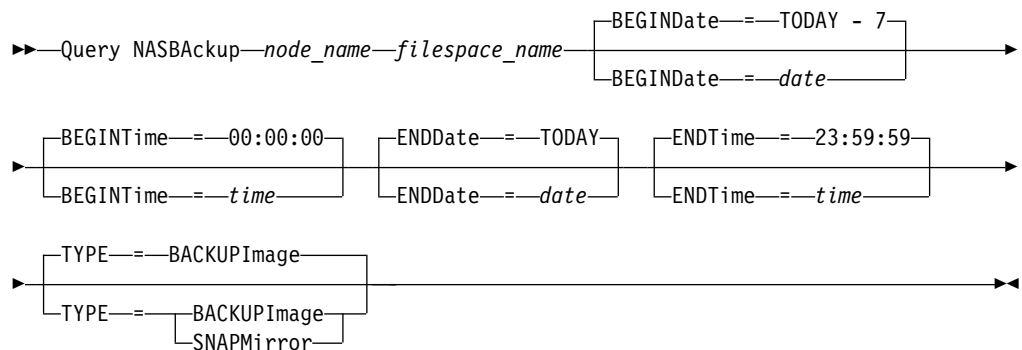
Use this command to display information about the file system image objects that have been backed up for a specific NAS node and file space. You can only use this command to display objects that were backed up for a NAS node using NDMP.

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.  To display information about the image objects that have been created a week ago, you can specify BEGINDATE=TODAY-7 or BEGINDATE= -7.

Value	Description	Example
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1 To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9 To include files that were active on the 10th day of the current month.

### 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 + <i>HH:MM</i>	The current time plus hours and minutes on the specified begin date	NOW+03:00 or +03:00. If you issue this command at 9:00 with BEGINTime=NOW+3 or BEGINTime=+3, the server displays image objects with a time of 12:00 or later on the begin date.
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes on the specified begin date	NOW-04:00 or -04:00. If you issue this command at 9:00 with BEGINTime=NOW-3:30 or BEGINTime=-3:30, the server 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

Value	Description	Example
TODAY- <i>days</i> or <i>-days</i>	The current date minus days specified. The maximum number of days you can specify is 9999.	TODAY-1 or -1.  To display information created up to yesterday, you can specify ENDDATE=TODAY-1 or simply ENDDATE= -1.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### 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.

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.  If you issue this command at 9:00 with ENDTIME=NOW+3:00 or ENDTIME= +3:00, the server displays image objects with a time of 12:00 or later 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.  If you issue this command at 9:00 with ENDTIME=NOW-3:30 or ENDTIME= -3:30, the server 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:

### BACKUPImage

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 filespace, `/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 filespace, `/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	MYPOOL
NAS2	vol/vol2	SnapMirror	1450.5	04/02/2008 11:03:21	STANDARD	MYPOOL

## Related commands

Table 241. 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.

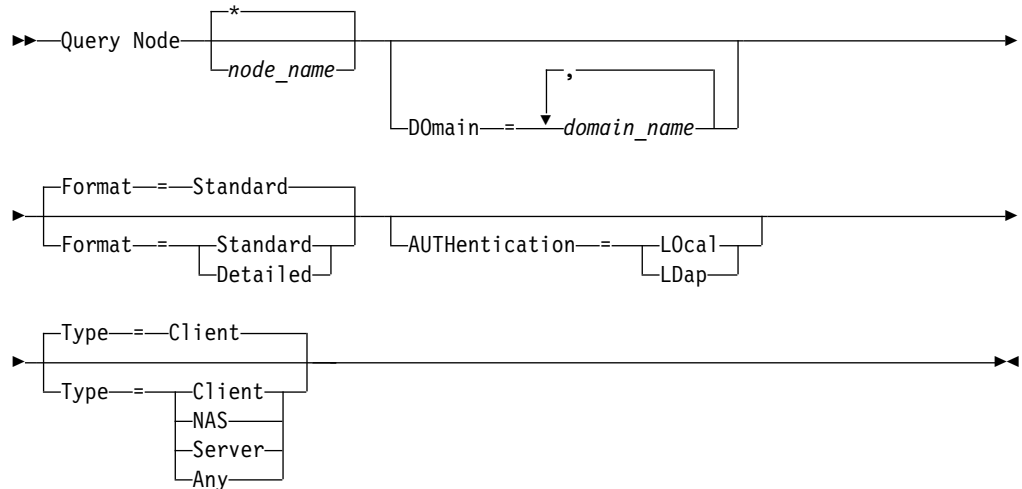
## QUERY NODE (Query nodes)

Use this command to view 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 that are 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. You can specify one of the following values:

##### **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. You can specify one of the following values:

**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 IBM Tivoli Storage Manager Tivoli Storage Manager servers.

**Authentication**  
Specifies the password authentication method for the node.

**Local**  
Display those nodes that authenticate to the Tivoli Storage Manager server.

**Ldap**  
Display those nodes that authenticate to an LDAP directory server. The node password is case-sensitive.

### Example: Display information about registered client nodes

Display information about 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 about 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 4, Level 0.0
Application Version: Version 6, Release 4, Level 0.4
Policy Domain Name: STANDARD
Last Access Date/Time: 09/24/2012 18:55:46
Days Since Last Access: 6

Password Set Date/Time: 09/24/2012 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: 09/24/2012 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: 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:
Email Address:
Deduplication: ServerOnly
Users allowed to back up: ALL
Replication State: Enabled
Replication Mode: Send
Backup Replication Rule: DEFAULT
Archive Replication Rule: ALL_DATA
Space Management Replication Rule: None
Replication Primary Server: PRODSERVER1
Last Replicated to Server: DRSERVER1
Client OS Name: WIN: Windows XP
Client Processor Architecture: x86
Client Products Installed: WIN, FCM, VE
Client Target Version: Version 6, Release 2, Level 0.0
Authentication: Local
SSLRequired: No
Split Large Objects: Yes
At-risk type: Default interval
At-risk interval:
Utility URL:
Replication Recovery of Damaged Files: Yes
Decommissioned:
Decommissioned Date:

```

## Field descriptions

### Node Name

The name of the client node.

### Platform

The operating system of the client node, 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 operating system type.

### Client OS Level

The level of the operating system for the client 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.

### Application Version

The version of the Tivoli Data Protection for VMware client.

### Policy Domain Name

The assigned policy domain 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 that elapsed since the last time that the client node accessed the server.

### Password Set Date/Time

The date and time that the password was set for the client node.

### Days Since Password Set

The number of days that elapsed since the password was set for the client node.

### Invalid Sign-on Count

The number of invalid sign-on attempts that were made since the last successful sign-on. This count can be non-zero only when the invalid password limit (**SET INVALIDPWLIMIT**) is greater than zero. When the number of invalid attempts equals the limit that is 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**

How long the most recent client node session lasted, in seconds.

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 running 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 URL of the Tivoli Storage Manager web client that is configured on the client system. You can use the URL in a web browser and in the Operations Center to remotely manage the client node.

**Node Type**

The type of client node. One of the following values is possible:

- 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 password expiration period of the client node.

**Keep Mount Point?**

Whether the client node retains a mount point during a session.

**Maximum Mount Points Allowed**

The number of mount points that a client node can 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 is 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 by using the **UPDATE NODE** command, 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. This evaluation might prevent the data store operations from acquiring 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 that is enabled. If the client has data validation that is 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 computer on which the node is located.

#### **Transaction Group Max**

Specifies the number of files per transaction committed that are transferred between a client and a server. Client performance might be improved by using a larger value for this option.

#### **Data Write Path**

Specifies the transfer path that is used when the client sends data to the server, storage agent, or both, during storage operations. 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 that is 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 following two options are available:

- 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.

**Tip:** If the node contains file spaces that are members of a file space collocation group, this field is left blank. You can find file space names by issuing the **QUERY FILESPACE** command.

**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.

**Email Address**

Specifies the email address of the client node.

**Deduplication**

The location where data is deduplicated. The value ServerOnly specifies that data stored by this node can be deduplicated on the server only. The Clientorserver value specifies that data stored by this node can be deduplicated on either the client or the server.

**Users allowed to back up**

Specifies whether a non-root user ID or only a root user ID can back up files to the server. ALL indicates all users, while ROOT indicates that just the root user ID can back up files to the server. This output is not available if the client node operating system is considered a single-user operating system.

**Replication State**

Indicates whether the node is enabled for replication. The following values are possible:

**Enabled**

The node is configured for replication and ready to replicate.

**Disabled**

The node is configured for replication but is not ready to replicate.

**None** The node is not configured for replication.

### **Replication Mode**

Indicates whether the node is configured as the source of or target for replicated data. If this field is blank, the node is not configured for replication. The following values are possible:

**Send** The node is configured as the source of data for replication.

### **Receive**

The node is configured as the target of data for replication.

### **SyncSend**

The data that belongs to the node is to be synchronized with the node data that is on the target replication server. Synchronization applies only to nodes whose data was imported from a source replication server and imported to the target replication server. Synchronization occurs during replication.

### **SyncReceive**

The data that belongs to the node is to be synchronized with the node data that is on the source replication server. Synchronization applies only to nodes whose data was imported from a source replication server and imported to the target replication server. Synchronization occurs during replication.

**None** The node is not configured for replication.

### **Replication Primary Server**

Specifies the source replication server for the client node.

### **Backup Replication Rule**

### **Archive Replication Rule**

### **Space Management Replication Rule**

The replication rule that applies to back up, archive, and space-managed data that belongs to the node. The following values are possible:

#### **ALL\_DATA**

Replicates backup, archive, or space-managed data. The data is replicated with normal priority.

#### **ACTIVE\_DATA**

Replicates active backup data. The data is replicated with normal priority.

**Attention:** If you specify **ACTIVE\_DATA** and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the **FORCERECONCILE=YES** parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

**ALL\_DATA\_HIGH\_PRIORITY**

Replicates backup, archive, or space-managed data. The data is replicated with high priority.

**ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the ACTIVE\_DATA replication rule except data is replicated with a high priority.

**DEFAULT**

Replicates backup, archive, or space-managed data according to the domain rule for the data type.

**NONE**

No data is replicated. For example, if the replication rule for archive data is NONE, archive data that belongs to the node is not replicated.

**Last Replicated to Server**

Specifies the name of the server that the node was last replicated to and the name of the server that the client fails over to during restore operations.

**Client OS Name**

The operating system of the client. The client deployment wizard uses this information to deploy a package to the client. This field is reported only for Tivoli Storage Manager clients at V6.2.0.0 and later.

**Client Processor Architecture**

The client architecture. The client deployment wizard uses this value to determine which package to deploy when the client is being updated. This field is reported only for Tivoli Storage Manager clients at V6.2.0.0 and later.

**Client Products Installed**

The products that are on the node. The following products might be listed:

- BA (Backup-Archive Client)
- VE (Virtual Environments)
- FCM (FlashCopy Manager)

**Client Target Version**

The version of the client that is installed at a time that is scheduled through the **DEFINE SCHEDULE** or **UPDATE SCHEDULE** command. This field is reported only for Tivoli Storage Manager clients at V6.2.0.0 and later.

**Authentication**

Specifies the password authentication method: LOCAL, LDAP, or LDAP (pending).

Authentication Target	Authentication Method
Tivoli Storage Manager server	LOCAL
LDAP directory server	LDAP
This node is configured to authenticate with an LDAP directory server, but the node did not yet authenticate.	LDAP (pending)

**SSL Required**

Specifies whether the security setting for the node requires Secure Sockets

Layer (SSL). Values can be YES, NO, or Default. You must have system level authority to update the node **SSLREQUIRED** setting.

### **Split Large Objects**

Specifies whether large objects that are stored by this node are automatically split into smaller pieces, by the server, to optimize server processing. Yes indicates that the server splits large objects (over 10 GB) into smaller pieces when stored by a client node. No indicates that this process is bypassed. The default value is Yes.

### **At-risk type**

Specifies the at-risk evaluation type. Values can be Default, Bypassed, or Custom. Default indicates that the node is evaluated with the same interval that was specified for the nodes classification by the **SET STATUSATRISKINTERVAL** command. Bypassed indicates that the node is not evaluated for at-risk status by the status monitor. Custom indicates that the node is evaluated with the interval that was specified by the **SET NODEATRISKINTERVAL** command, rather than the interval that was specified by the **SET STATUSATRISKINTERVAL** command.

### **At-risk interval**

Specifies the number of hours between two client backup activities, or two replication activities, after which the status monitor indicates that the activity is at risk. This field contains a value only when the At-risk type field contains the value of Custom.

### **Utility URL**

Specifies the address of the Tivoli Storage Manager client management services that are configured on the client system. This URL is used by the Operations Center to access client log files so that you can remotely diagnose client issues from the Operations Center.

### **Replication Recovery of Damaged Files**

Specifies whether damaged files can be recovered for this node from a target replication server.

### **Decommissioned**

Specifies whether the client node is decommissioned. The following values are possible:

**YES** Specifies that the node is decommissioned.

#### **Null value**

Specifies that the node is not decommissioned.

#### **PENDING**

Specifies that the node is being decommissioned, or the decommission process failed.

**Tip:** If you want to determine the status of a pending decommission process, follow the instructions in Decommissioning a client node.

### **Decommissioned Date**

Specifies the date that the client node was decommissioned.

### **Example: Display information about node roles**

The example output is only a portion of the full display.

```
query node alvin f=d
```



```
Proxynode Agent:
  Node Groups:
  Email Address:
  Deduplication: ServerOnly
Users allowed to back up: All
  Role: Server
  Role Override: UseReported
Processor Vendor: ORACLE
Processor Brand: UltraSPARC-T2
Processor Type: 4
Processor Model:
Processor Count: 1
Hypervisor:
API Application: NO
Scan Error: NO
MAC Address:
```

## Field Descriptions

**Role** The processor role as reported by the client.

### Role Override

The override value for role, which is specified with the **UPDATE NODE** command.

### Processor Vendor

The processor vendor as reported by the client.

### Processor Brand

The processor brand as reported by the client.

### Processor Type

The processor type as reported by the client. This value specifies the number of processor cores that are used for PVU calculation.

### Processor Model

The processor model as reported by the client.

### Processor Count

The processor count as reported by the client.

### Hypervisor

The hypervisor as reported by the client.

### API Application

The client indicator that the client is an API application.

### Scan Error

The indicator of whether the latest scan for processor information might be failing and needs investigation.

### MAC Address

MAC Address as reported by the client.

## Example: View all nodes that authenticate to the Tivoli Storage Manager server

If you want to view all nodes that authenticate locally, specify the following command:

```
query node * authentication=local
```

Node Name	Platform	Policy Domain Name	Days Since Last Access	Days Since Password Set	Locked?
NODE1	WinNT	STANDARD	3	3	No
LOCAL	(?)	STANDARD	7	7	No

## Related commands

Table 242. Commands related to QUERY NODE

Command	Description
LOCK NODE	Prevents a client from accessing the server.
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.
QUERY REPLNODE	Displays information about the replication status of a client node.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REGISTER NODE	Defines a client node 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.
REMOVE REPLNODE	Removes a node from replication.
RENAME NODE	Changes the name for a client node.
REPLICATE NODE	Replicates data in file spaces that belong to 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 that are associated with a client node.

## QUERY NODEDATA (Query client data in volumes)

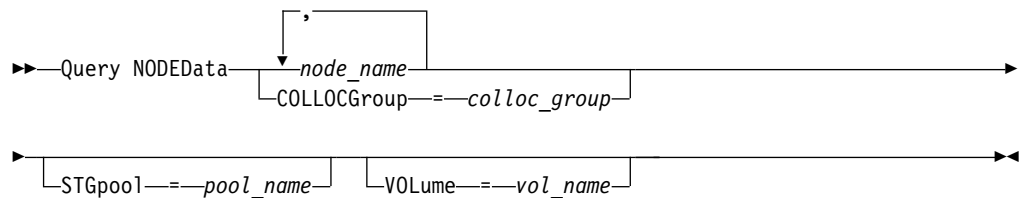
Use this command to display information about the data for one or more nodes in a sequential access storage pool. **QUERY NODEDATA** displays the name of the volume on which a node's data is written and the amount of space that is occupied by the data on that volume. This information is useful when you determine how to group nodes into collocated storage pools.

### Privilege class

**Restriction:** You cannot use this command to display information for container storage pools.

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 to specify 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.

**Important:** If the amount of space that is needed to complete the query about a collocation group exceeds the SQL buffer limit, the **QUERY NODEDATA** command can fail. If the command fails for this reason, issue the **QUERY COLLOCGROUP** command to display a list of nodes in the group. Then, issue the **QUERY NODEDATA** command for each node in the group.

#### **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 is 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.

## 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” for field descriptions.

```
query nodedata e*
```

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

## 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.

```
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

If you specify a file space collocation group, only the volumes of the file spaces that belong to the collocation group are displayed. If you specify a file space collocation group and a volume, the file space volumes within the collocation group that are also in the specified volume are displayed.

## 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 that is occupied by the node's data. Physical space includes empty space within aggregates, from which files might be deleted or expired.

## Related commands

Table 243. Commands related to **QUERY NODEDATA**

Command	Description
DEFINE COLLOGROUP	Defines a collocation group.
DEFINE COLLOCMEMBER	Adds a client node or file space 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 or file space 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 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.
UPDATE STGPOOL	Changes the attributes of a storage pool.

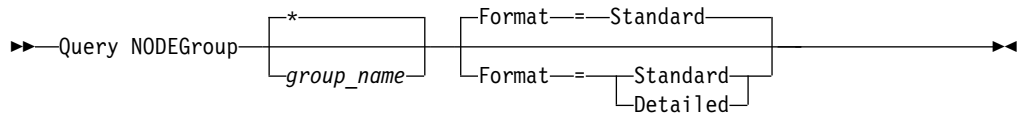
## QUERY NODEGROUP (Query a node group)

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 833 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 833 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 244. 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.

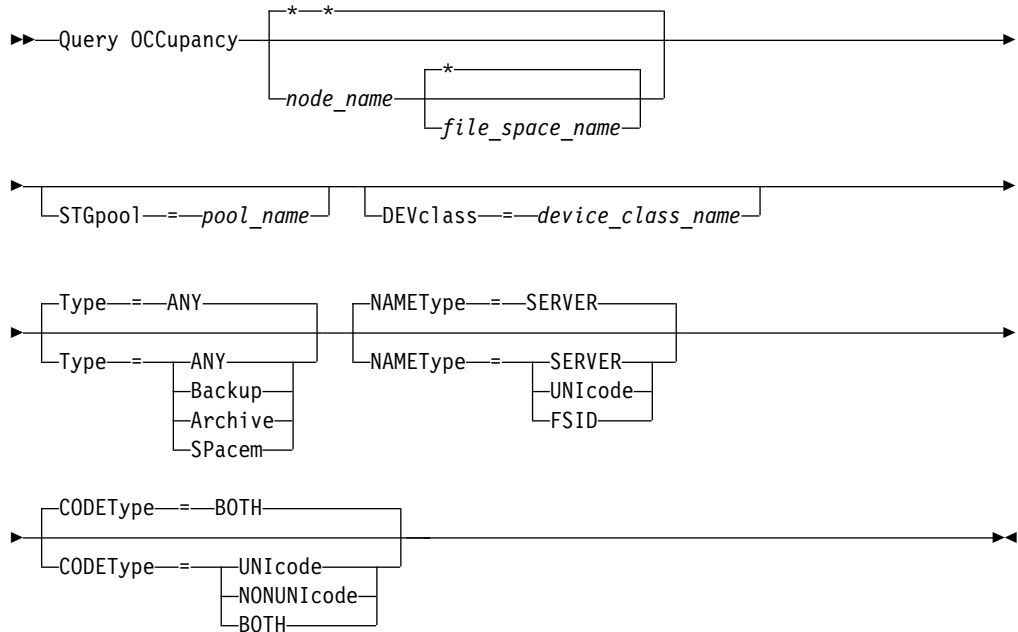
## QUERY OCCUPANCY (Query client file spaces in storage pools)

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 might need to have the server convert the file space name that you enter. For example, you might need to have the server convert the name that 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.

#### **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 that is 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 that are 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: back up versions of files, archived copies of files, and files that are 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 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 file spaces that are only Unicode enabled.

**NONUNICODE**

Include file spaces that are not only Unicode enabled.

## BOTH

Include file spaces regardless of code page type.

### Example: Display file spaces assigned to a specific node

Display information about 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 about 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. If the node was previously deleted, the node name DELETED is displayed.

**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.

If the file space was previously deleted, the file space name DELETED is displayed.

File space names can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion

depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

**Storage Pool Name**

The storage pool where the file space is located.

**Number of Files**

The number of logical files that belong to the file space and are stored in this storage pool. When storing a file larger than 10 GB, the server splits the file into 10 GB fragments. The number of fragments is also included in this value for occupancy calculations.

**Physical Space Occupied (MB)**

The amount of physical space that is occupied by the file space. Physical space includes empty space within aggregates, from which files might have been deleted or expired. For this value, 1 MB = 1048576 bytes.

**Tip:** This field does not display a value for storage pools that are set up for data deduplication. If you turn off data 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 that is occupied by logical files in the file space. Logical space is the space that is actually used to store files, excluding empty space within aggregates. For this value, 1 MB = 1048576 bytes.

**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 245. Commands related to QUERY OCCUPANCY*

Command	Description
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
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)

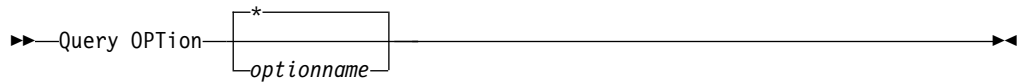
Use this command to display information about server options.

Change server options by editing the server options file or by issuing the **SETOPT** command. When 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 character

View the option settings for all options that begin with L.

```
query option l*
```

Server Option	Option Setting
-----	-----
Language	AMENG

### Example: Display LDAP directory servers

View the settings for all LDAP directory servers.

```
query option ldapurl
```

Server Option	Option Setting
-----	-----
LDAP URL	ldap://tophoy.tucson.com/cn=tsmdata
LDAP URL	ldap://krypton.ibm.com/ou=tsmdata,dc=ibm,dc=com

### Field descriptions

#### Server Option

Specifies the name of the option in the server options file.

### Option Setting

Specifies the name of the option in the server options file.

### Related commands

*Table 246. Commands related to QUERY OPTION*

Command	Description
SETOPT	Updates a server option without stopping and restarting the server.

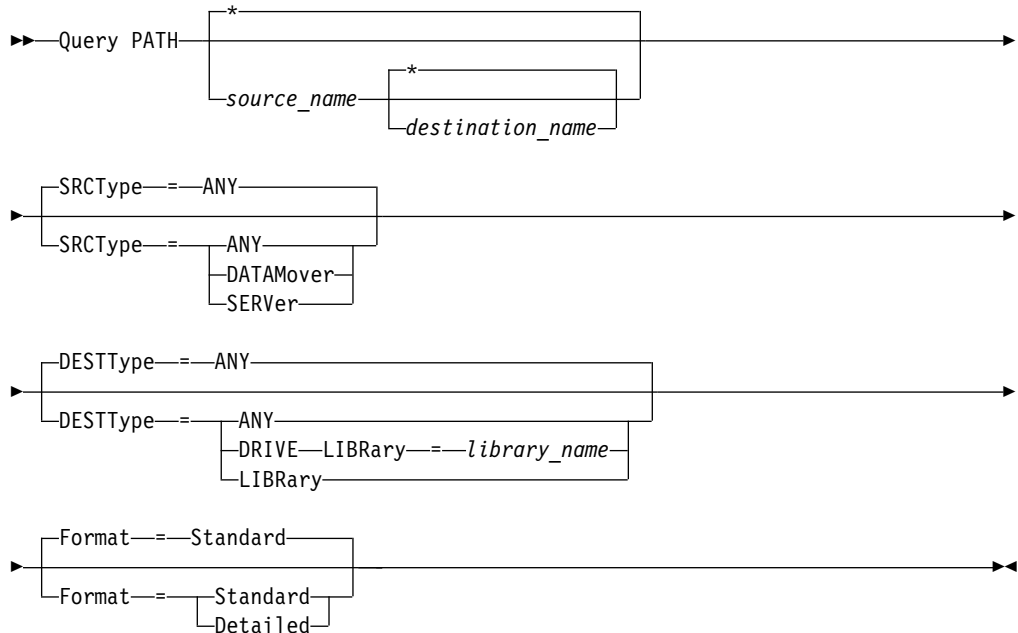
## QUERY PATH (Display a path definition)

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 842 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 842 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.

### Device

The name of the device that is the destination.

### External Manager

The name of the external manager.

### ZOS Media Server

The name of the z/OS media server.

### Comm. Method

Specifies the type of communication method.

**LUN** Specifies the logical unit name through which the disk can be accessed by the source.

### Initiator

Specifies the initiator of the communication.

### Directory

Specifies the directory location of a file on the source.



**On-Line**

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 247. Commands related to QUERY PATH*

<b>Command</b>	<b>Description</b>
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.

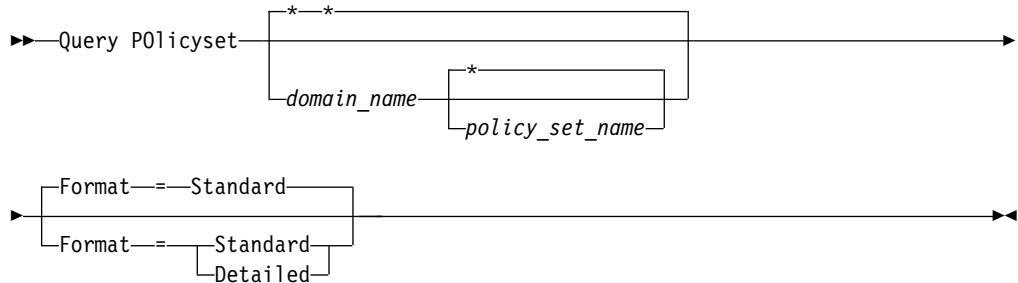
## QUERY POLICYSET (Query a policy set)

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 845 for field descriptions.

```
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 policysset 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 248. Commands related to QUERY POLICYSET*

<b>Command</b>	<b>Description</b>
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 about active background processes.

To cancel background processes, issue the **CANCEL PROCESS** command. To display detailed information about node replication processes, issue the **QUERY REPLICATION** command.

## Privilege class

Any administrator can issue this command.

## Syntax



## Parameters

### *process\_number*

Specifies the number of the background process to be queried. This parameter is optional. If not specified, information about all background processes is displayed.

## Example: Query a single background process

Display information about background process 202. See “Field descriptions” on page 849 for field descriptions.

`query process 202`

Process Number	Process Description	Process Status
202	EXPORT SERVER	ANR0NNNI EXPORT Identifier MYEXPORTSERVER ANR0648I Have copied the following: 8 Domains 2 Policy Sets 10 Management Classes 4 Copy Groups 1 Administrators 746 Bytes (0 errors have been detected) Current input volume(s): C:\BUILD\540\ GA\BUILD\NT\I386\DEBUG\ -00000014.BFS,(6 Seconds)

## Example: Query all background processes

Display information about all background processes. See “Field descriptions” on page 849 for field descriptions.

`query process`

Process Number	Process Description	Process Status
304	IDENTIFY DUPLICATES	Storage Pool FILEPOOL, Volume /tspool2/00006664. BFS, Files Processed: 2000, Duplicate Extents Found: 344, Duplicate Bytes Found: 3,238,123, Current Physical File (bytes): 2,626,676,296. Status: Processing
284	IDENTIFY DUPLICATES	Storage Pool FILEPOOL, Volume /tspool2/00006666. BFS, Files Processed: 2000, Duplicate Extents Found: 344, Duplicate Bytes Found: 3,238,123, Current Physical File (bytes): None. Status: Idle
4	Replicate Node	Replicating Node(s) IRONMAN. File spaces complete: 0. File spaces identifying and replicating: 1. File spaces replicating: 0. File spaces not started: 3. Files current: 11,920. Files replicated: 0 of 0. Files updated: 0 of 0. Files deleted: 0 of 0. Amount Replicated: 11,482 KB of 11,482 KB. Amount transferred: 11,482 KB. Elapsed time: 0 Day(s), 0 Hour(s), 1 Minute(s).
37	Expiration	Processed 12 nodes out of 30 total nodes, examined 411 objects, deleting 411 backup objects, 0 archive objects, 0 DB backup volumes, 0 recovery plan files; 0 objects have been retried and 0 errors encountered.

### Example: Verify that a replication recovery process was initiated

After you start a node replication process with file recovery enabled, verify that the target replication server initiated the file recovery process. Issue the **QUERY PROCESS** command on the target replication server. For descriptions of fields, see “Field descriptions” on page 849.

query process

Process Number	Process Description	Process Status
4	Replicate Node - Recovery.	Replicating node(s) 3MAUTOIMPORT. File spaces complete: 87. File spaces identifying and replicating: 0. File spaces replicating: 6. File spaces not started: 0. Files current: 0. Files replicated: 0 of 14. Files updated: 0 of 0. Files deleted: 0 of 0. Amount replicated: 0 KB of 11,688 bytes. Amount transferred: 0 KB. Elapsed time: 0 Day(s), 0 Hour(s), 1 Minute(s).

**Example: Verify that damaged files are being recovered during a replication process**

After you start a node replication process with file recovery enabled, verify that damaged files are being recovered. Issue the **QUERY PROCESS** command on the source replication server. For descriptions of fields, see "Field descriptions."

query process

Process Number	Process Description	Process Status
6	Replicate Node ( As Secondary Recovery )	Recovering damaged files from server SERVER2, process 4, number of active sessions 10.

**Field descriptions**

**Process Number**  
Specifies the number that is assigned to the active background process.

**Process Description**  
Specifies a description of the active background process.

**Process Status**  
Specifies the status of the active background process.

**Tip:** When a node replication process is finished on the target replication server, only end process information is stored in the activity summary table. The full summary for the replication process is stored in the activity summary table on the source replication server.

**Related commands**

Table 249. Command related to **QUERY PROCESS**

Command	Description
CANCEL EXPORT	Deletes a suspended export operation.
CANCEL PROCESS	Cancels a background server process.
IDENTIFY DUPLICATES	Identifies duplicate data in a storage pool.

Table 249. Command related to **QUERY PROCESS** (continued)

<b>Command</b>	<b>Description</b>
QUERY EXPORT	Displays the export operations that are currently running or suspended.
QUERY REPLICATION	Displays information about node replication processes.
QUERY REPLNODE	Displays information about the replication status of a client node.
RESTART EXPORT	Restarts a suspended export operation.
SUSPEND EXPORT	Suspends a running export operation.



## QUERY PROFILE (Query a profile)

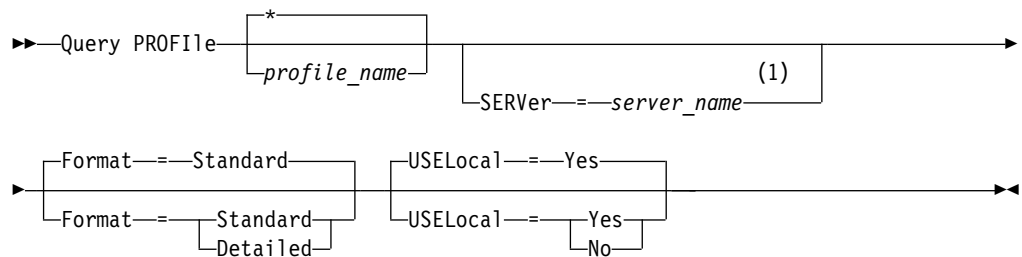
Use this command to display information about profiles and associated objects. Issue this command from a configuration manager or from a managed server. You can use this command to get profile information from any configuration manager defined to the server, even if the server does not subscribe to any 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.

**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 250. Commands related to QUERY PROFILE*

<b>Command</b>	<b>Description</b>
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.

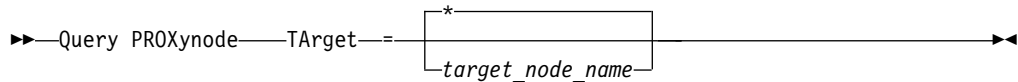
## QUERY PROXYNODE (Query proxy authority for a client node)

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

### Field descriptions

#### Target Node

Specifies the name of the node targeted by the node with proxy authority.

#### Agent Node

Specifies the name of the agent node.

### Related commands

Table 251. 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 PVUESTIMATE (Display processor value unit estimate)

Use this command to obtain an estimate of the client devices and server devices that are being managed by the Tivoli Storage Manager server. In addition, this command provides an estimate of the processor value unit (PVU) totals for the server devices.

This command generates a PVU estimate that is based on the number of logical nodes that are defined to the Tivoli Storage Manager server. By contrast, the calculation of license obligations is based on the number of physical computers. There might not be a one-to-one correlation between the number of logical nodes and the number of physical computers. The report that is generated by the **QUERY PVUESTIMATE** command is an estimate, which is not legally binding.

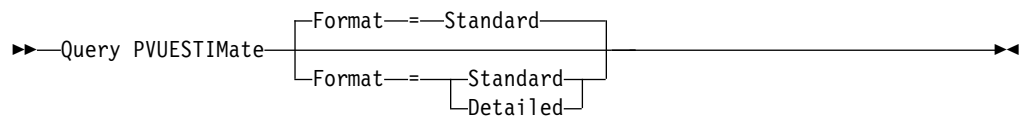
For purposes of the **QUERY PVUESTIMATE** command, nodes on Microsoft Windows 7, Microsoft Windows XP Professional, and Apple systems are assumed to be client devices. Nodes on all other platforms are considered to be server devices. The server on which Tivoli Storage Manager is running is also classified as a server device. However, you can reclassify server devices as client devices if required. If your system includes retired workstations, test workstations, or others that can be ignored for purposes of PVU calculation, you can specify them as type other. To change a node classification, use the **UPDATE NODE** command or the **REGISTER NODE** command.

**Note:** The PVU information reported by Tivoli Storage Manager is not considered an acceptable substitute for the IBM License Metric Tool.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### Format

Specifies the output format. This parameter is optional. The default is Standard. The following values can be used:

#### Standard

Specifies standard output.

#### Detailed

Specifies detailed output.

### Example: Display the estimated number of devices and PVU

Display the estimated number of client devices and server devices, and the estimated PVU for the server devices, for a Tivoli Storage Manager server. Issue the following command:

```
query pvuestimate
```

Table 252. Sample output for several products managed by one Tivoli Storage Manager server

Product	Number of Client Devices	Number of Server Devices	PVU of Server Devices
Tivoli Storage Manager Extended Edition	1,000	905	90,500
Tivoli Storage Manager for Storage Area Networks	50	10	1,000
Tivoli Storage Manager for Space Management	0	0	0
Tivoli Storage Manager for Mail	0	25	5,000
Tivoli Storage Manager for Databases	0	1,025	20,500
Tivoli Storage Manager for Enterprise Resource Planning	0	25	5,000
Tivoli Storage Manager for System Backup and Recovery	0	0	0

Other Node Classifications	Number
Nodes earlier than Version 6.3 with no PVU information available at this time	10
Nodes at Version 6.3 or later with no PVU match	9
Nodes classified by the administrator as "other-device"	8
Nodes defined as a non-licensed API application	6

The following list provides details about the example fields:

**Product**

The Tivoli Storage Manager product name.

**Number of Client Devices**

The estimated number of client devices that are managed by the product. By default, only nodes on Microsoft Windows 7, Microsoft Windows XP Professional, and Apple systems are assumed to be client devices.

**Number of Server Devices**

The estimated number of server devices that are managed by the product. By default, nodes on all platforms except for Microsoft Windows 7, Microsoft Windows XP Professional, and Apple systems are assumed to be server devices. This number also includes the server on which Tivoli Storage Manager is running.

**PVU of Server Devices**

The estimated PVUs of all nodes that are connected as server devices.

**Nodes earlier than Version 6.3 with no PVU information available at this time**

Devices that do not report processor information to the server.

**Nodes at Version 6.3 or later with no PVU match**

Devices that do not report all required values or some values were reported as "Unknown".

**Nodes classified by the administrator as "other-device"**

Nodes that are excluded from PVU counting by the administrator by using the `update node roleoverride=other` command.

**Nodes defined as a non-licensed API application**

Nodes such as DB2 backup or custom API applications.

## Example: Display detailed node information

Display information for individual nodes by specifying the detailed (d) value for the **Format** parameter. Issue the following command:

```
tsm: PATMOS_630> query pvestimate f=d
```

Table 253. Node classifications for specific products

Product	Number of Client Devices	Number of Server Devices	PVU of Server Devices
Tivoli Storage Manager Extended Edition	1,000	905	90,500
- banode1	1		
- banode2		1	200
- banode3	1		
- banode3		1	100
Tivoli Storage Manager for Storage Area Networks	50	10	1,000
- stagent1		1	50
- stagent2		1	100
Tivoli Storage Manager for Space Management	0	0	0
Tivoli Storage Manager for Mail	0	25	5,000
- mailnode1		1	200
- mailnode2		1	100
Tivoli Storage Manager for Databases	0	1,025	20,500
- dbnode1		1	200
- dbnode2		1	100
Tivoli Storage Manager for Enterprise Resource Planning	0	25	5,000
- erpnode1		1	50
- erpnode2		1	100
Tivoli Storage Manager for System Backup and Recovery	0	0	0
<b>Other Node Classifications</b>			<b>Number</b>
Nodes earlier than Version 6.3 with no PVU information available at this time			10
- oldnode1			1
- oldnode2			1
- mailnote44			1
- erpnode66			1
Nodes at Version 6.3 or later with no PVU match			10
- badcitnode1			1

<b>Other Node Classifications</b>	<b>Number</b>
- badcitnode2	1
- mailnode23	1
- erpnode34	1
<hr/>	
Nodes classified by administrator as "other-device"	8
- overriddennode1	1
- overriddennode2	1
- mailnode77	
<hr/>	
Nodes defined as a non-licensed API application	6
- vendorapinode1	1
- vendorapinode2	1

## **Related commands**

*Table 254. Commands related to QUERY PVUESTIMATE*

<b>Command</b>	<b>Description</b>
AUDIT LICENSES	Verifies compliance with defined licenses.
QUERY LICENSE	Displays information about licenses and audits.
QUERY NODE	Displays partial or complete information about one or more clients.
REGISTER LICENSE	Registers a license with the Tivoli Storage Manager server.
REGISTER NODE	Defines a client node to the server and sets options for that user.
SET CPUINFOREFRESH	Specifies the number of days between client scans for workstation information used for PVU estimates.
SET LICENSEAUDITPERIOD	Specifies the number of days between automatic license audits.
UPDATE NODE	Changes the attributes that are associated with a client node.



## QUERY RECOVERYMEDIA (Query recovery media)

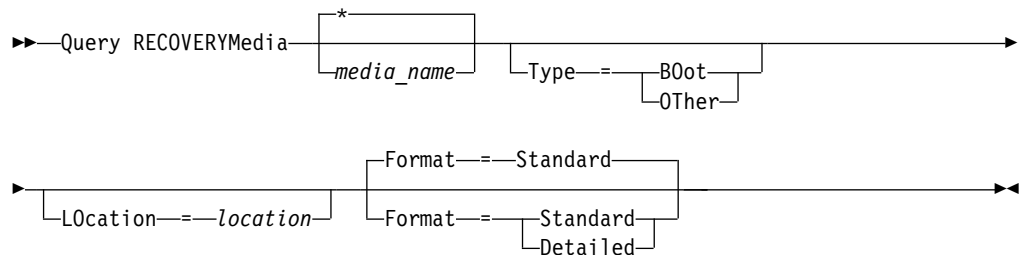
Use this command to display information about the media (for example, boot media) needed to recover a machine. Media are displayed in alphabetical order by name.

**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.

##### **0ther**

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 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.

## Related commands

*Table 255. Commands related to QUERY RECOVERYMEDIA*

<b>Command</b>	<b>Description</b>
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 REPLICATION (Query node replication processes)

Use this command to display information about running and ended node-replication processes.

Issue this command on the server that acts as a source for replicated data.

**Important:** You cannot display information about running replication processes for client nodes that are being converted from import and export operations to replication operations. The conversion process might run for a long time, but it occurs only once for a client node that is being converted.

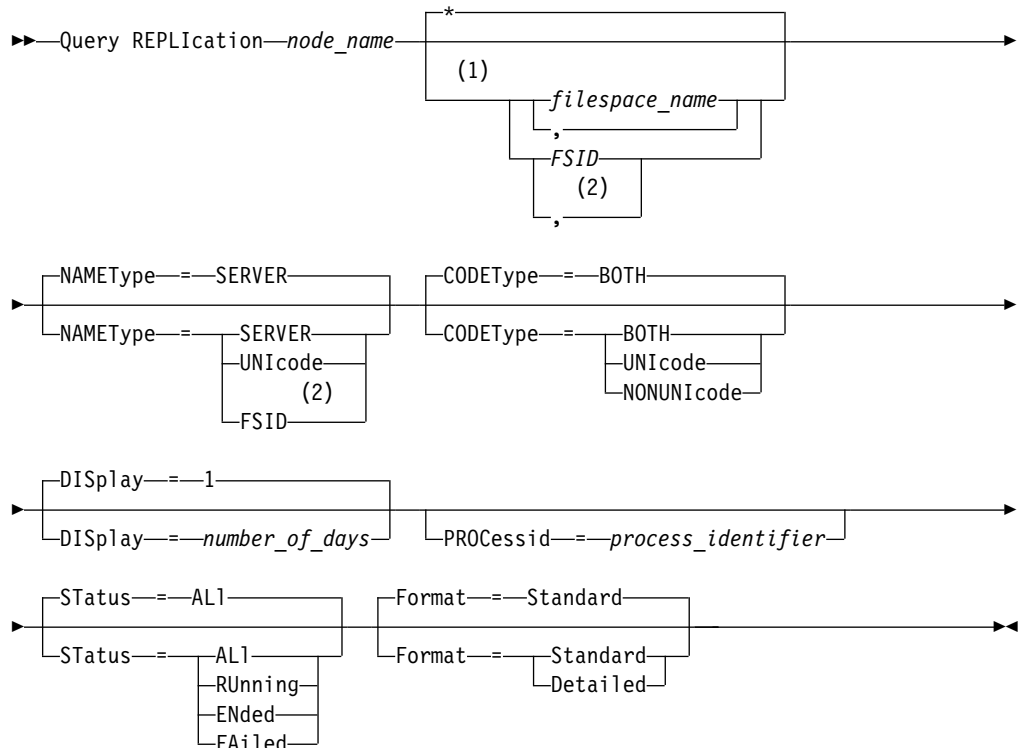
By default, records about completed node-replication processes are retained for 30 calendar days. A *calendar day* consists of 24-hours, from midnight to midnight.

To display the retention period, issue the **QUERY STATUS** command. Check the value in the **Replication Record Retention Period** field. To change the retention period, issue the **SET REPRETENTION** command.

### Privilege class

Any administrator can issue this command.

### Syntax



### Notes:

- 1 Do not mix FSIDs (file space identifiers) and file space names in the same command.
- 2 Do not specify FSID if you use wildcard characters for the client node name.

## Parameters

### *node\_name* (Required)

Specifies the name of the client node to be queried. You can use wildcard characters when you specify this name, with one exception. If the value of the **NAMETYPE** parameter is FSID, do not specify wildcard characters for the client node name. The FSID value indicates the file space identifier. File spaces with identical names can have different identifiers in different client nodes.

### *file\_space\_name* or *FSID*

Specifies the name of the file space or the file space identifier (FSID) to be queried. A name or FSID is optional. If you do not specify a name or an FSID, all file spaces are queried.

### *file\_space\_name*

Specifies the name of the file space that has data to be queried. File space names are case-sensitive. To determine the correct capitalization for the file space, issue the **QUERY FILESPACE** command. Separate multiple names with commas with no intervening spaces. When you specify a name, you can use wildcard characters.

A server that has clients with Unicode-enabled file spaces might have to convert the file space name. For example, the server might have to convert a name from the server code page to Unicode. For details, see the **NAMETYPE** parameter. If you do not specify a file space name, or if you 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.

### *FSID*

Specifies the file space identifier for the file space to be queried. The server uses FSIDs to find the file spaces to replicate. To determine the FSID for a file space, issue the **QUERY FILESPACE** command. Separate multiple FSIDs with commas with no intervening spaces. If you specify an FSID, the value of the **NAMETYPE** parameter must be FSID.

## **NAMETYPE**

Specifies how you want the server to interpret the file space names that you enter. You can use this parameter for Tivoli Storage Manager clients that are Unicode-enabled and that have Windows, Macintosh OS X, or NetWare operating systems.

Use this parameter only if you enter a partly qualified or fully qualified file space name. The default value is SERVER. You can specify one of the following values:

### **SERVER**

The server uses the server code page to interpret file space names.

### **UNICODE**

The server converts file space names from the server code page to the UTF-8 code page. The success of the conversion depends on the characters in the name and the server code page. Conversion can fail if the string includes characters that are not available in the server code page.

Conversion can also fail if the server cannot access system conversion routines.

### **FSID**

The server interprets file space names by using their file space identifiers.

**CODEType**

Specifies the type of file spaces to be included in the query. The default value is **BOTH**, which means that file spaces are included regardless of code page type. Use this parameter only if you enter a single wildcard character for the file space name. You can specify one of the following values:

**UNICODE**

Include file spaces that are in Unicode only.

**NONUNICODE**

Include file spaces that are not in Unicode only.

**BOTH**

Include all file spaces regardless of code page type.

**DISplay**

Specifies the number of days of node replication history to display. The default value is 1, which displays information about running node replication processes and about processes that completed during the current calendar day. The maximum value is 9999.

You can specify a number that is the same as or less than the number of days that are specified as the retention period for the replication history records. If you specify a value that is more than the value of the replication retention period or more than the number of days that replication records are collected, the server displays only the number of replication history records that are available. For example, suppose that the replication retention period is 30 days and that the replication process is running for only 10 days. If you specify **DISPLAY=20**, only 10 days of replication history are displayed.

**PROcessid**

Specifies the node replication history that is associated with a particular process identified by the process identifier. This parameter is optional. If you do not specify this parameter, all processes are displayed for the number of days that are specified by the **DISPLAY** parameter.

Restarting the server can cause the server to reuse process IDs. Reuse of process IDs can result in duplicate process IDs for separate processes.

**STatus**

Specifies the status of the file spaces to query. This parameter is optional. The default value is **ALL**. You can specify one of the following values:

**ALL**

Specifies all file spaces that are replicating, file spaces that replicated successfully, and file spaces that did not finish replicating or replicated with errors.

**RUnning**

Specifies all file spaces that are replicating to the target replication server.

**ENded**

Specifies all file spaces that replicated successfully and file spaces that did not finish replicating or replicated with errors.

**FAiled**

Specifies all file spaces that did not finish replicating or replicated with errors.

**Format**

Specifies how the information is displayed. This parameter is optional. The default value is STANDARD. You can specify one of the following values:

**Standard**

Specifies that partial information is displayed for node replication processes.

**Detailed**

Specifies that all available information for the node replication processes is displayed.

**Example: Display information about replication processes for a file space**

Display information about replication processes for a file space in client node PAYROLL. The file space identifier is 10.

```
query replication ironman
```

NodeName	Filespace Name	FSID	Start Time	End Time	Status	Phase
IRONMAN	/space	2	02/08/11 21:44:19	02/08/11 21:48:14	Ended	None

```
query replication ironman format=detailed
```

```

Node Name: IRONMAN
Filespace Name: /space
FSID: 2
Start Time: 02/08/11 21:44:19
End Time: 02/08/11 21:48:14
Status: Ended
Process Number: 4
Command: replicate node ironman
Phase: None
Process Running Time: 0 Day(s) 0 Hour(s)
4 Minute(s)
Completion State: Complete
Reason For Incompletion: None
Backup Last Update Date/Time:
Backup Target Server:
Backup Files Needing No Action: 0
Backup Files To Replicate: 0
Backup Files Replicated: 0
Backup Files Not Replicated Due to Errors: 0
Backup Files Not Yet Replicated: 0
Backup Files To Delete: 0
Backup Files Deleted: 0
Backup Files Not Deleted Due To Errors: 0
Backup Files To Update: 0
Backup Files Updated: 0
Backup Files Not Updated Due To Errors: 0
Backup Bytes To Replicate (MB): 0
Backup Bytes Replicated (MB): 0
Backup Bytes Transferred (MB): 0
Backup Bytes Not Replicated Due
To Errors (MB): 0
Backup Bytes Not Yet Replicated (MB): 0

Archive Last Update Date/Time: 02/08/11 21:48:14
Archive Target Server: NIGLINA
Archive Files Needing No Action: 0
Archive Files To Replicate: 39,416
Archive Files Replicated: 39,206
Archive Files Not Replicated Due to Errors: 210
Archive Files Not Yet Replicated: 0
Archive Files To Delete: 0
Archive Files Deleted: 0
Archive Files Not Deleted Due To Errors: 0
Archive Files To Update: 0
Archive Files Updated: 0

```



```

Archive Files Not Updated Due To Errors: 0
  Archive Bytes To Replicate (MB): 4,335
  Archive Bytes Replicated (MB): 4,335
  Archive Bytes Transferred (MB): 0
  Archive Bytes Not Replicated
    Due To Errors (MB): 0
  Archive Bytes Not Yet Replicated (MB): 0

  Space Managed Last Update Date/Time:
  Space Management Target Server:
Space Managed Files Needing No Action: 0
  Space Managed Files To Replicate: 0
  Space Managed Files Replicated: 0
  Space Managed Files Not Replicated
    Due to Errors: 0
Space Managed Files Not Yet Replicated: 0
  Space Managed Files To Delete: 0
  Space Managed Files Deleted: 0
  Space Managed Files Not Deleted
    Due To Errors: 0
  Space Managed Files To Update: 0
  Space Managed Files Updated: 0
  Space Managed Files Not Updated
    Due To Errors: 0
Space Managed Bytes To Replicate (MB): 0
  Space Managed Bytes Replicated (MB): 0
  Space Managed Bytes Transferred (MB): 0
  Space Managed Bytes Not Replicated
    Due To Errors (MB): 0
Space Managed Bytes Not Yet Replicated (MB): 0
  Total Files Needing No Action: 0
  Total Files To Replicate: 39,416
  Total Files Replicated: 39,206
Total Files Not Replicated Due To Errors: 210
  Total Files Not Yet Replicated: 0
  Total Files To Delete: 0
  Total Files Deleted: 0
  Total Files Not Deleted Due To Errors: 0
  Total Files To Update: 0
  Total Files Updated: 0
Total Files Not Updated Due To Errors: 0
  Total Bytes To Replicate (MB): 4,335
  Total Bytes Replicated (MB): 4,335
  Total Bytes Transferred (MB):
  Total Bytes Not Replicated
    Due to Errors (MB):
  Total Bytes Not Yet Replicated (MB):
  Estimated Percentage Complete: 100
  Estimated Time Remaining:
  Estimated Time of Completion:

```

## Field descriptions

### Node Name

The name of the client node whose data is displayed.

### Filespace Name

The name of the client file space whose data is displayed.

**FSID** The file space identifier.

### Start Time

The date and time that the node replication process started.

### End Time

The date and time that the node replication process ended.

**Status** The status of the node replication process. The following values are possible:

**Running**

The process is active and is either searching for eligible data or sending data to the target replication server.

**Ended** The process ended or failed.

**Failed** The process failed.

**Process Number**

The identifier for the node replication process.

The same process number can have different start times. If a replication process starts and the server is restarted, the server begins assigning process numbers that begin with the number 1. Replication processes that start after a server restart can obtain process numbers that are already assigned to other replication processes in the replication history. To identify unique replication processes, use the start time.

**Command**

The command that was issued to start the node replication process.

**Phase** The phase of a running node-replication process. The following phases are listed in the order in which they occur:

**Identifying**

The node replication process started to identify data to be replicated, but data is not yet being sent to the target replication server.

**Identifying and replicating**

The node replication process is identifying data to be replicated and transferring the data to the target replication server.

**Replicating**

The node replication process identified the data and is transferring files to the target replication server.

**None** The node replication process is not running.

**Process Running Time**

The running time of the node replication process.

**Completion State**

The state of the node replication process. The following values are possible:

**Complete**

The node replication process completed.

**Incomplete**

The node replication process ended without running to completion. To determine the reason, check the value in the Reason for Incompletion field.

**Reason for Incompletion**

The reason why the node replication process ended without completing. Possible values include *cancelled* and *other*. The value *other* can indicate that the server was halted during replication or that the server failed.

**Backup Last Update Date/Time**

The date and time that statistics for backup were last updated. The specified time is the time that the files in the file space were identified for replication or when each batch of files was sent to the target replication server.

**Archive Last Update Date/Time**

The date and time that statistics for archive were last updated. The specified time is the time that the files in the file space were identified for replication or when each batch of files was sent to the target replication server.

**Space Managed Last Update Date/Time**

The date and time that statistics for space-managed files were last updated. The specified time is the time that the files in the file space were identified for replication or when each batch of files was sent to the target replication server.

**Backup Target Server**

The name of the target replication server for backup files.

**Archive Target Server**

The name of the target replication server for archive files.

**Space Management Target Server**

The name of the target replication server for space-managed files.

**Backup Files Needing No Action**

The number of backup files in the file space that did not need to be replicated, updated, or deleted.

**Archive Files Needing No Action**

The number of archive files in the file space that did not need to be replicated, updated, or deleted.

**Space Managed Files Needing No Action**

The number of space-managed files in the file space that did not need to be replicated, updated, or deleted.

**Backup Files To Replicate**

The number of backup files to replicate to the target replication server.

**Archive Files To Replicate**

The number of archive files to replicate to the target replication server.

**Space Managed Files To Replicate**

The number of space-managed files to replicate to the target replication server.

**Backup Files Replicated**

The number of backup files that are replicated to the target replication server.

**Archive Files Replicated**

The number of archive files that are replicated to the target replication server.

**Space Managed Files Replicated**

The number of space-managed files that are replicated to the target replication server.

**Backup Files Not Replicated Due To Errors**

The number of backup files that were not replicated to the target replication server because of errors.

**Archive Files Not Replicated Due To Errors**

The number of archive files that were not replicated to the target replication server because of errors.

**Space Managed Files Not Replicated Due To Errors**

The number of space-managed files that were not replicated to the target replication server because of errors.

**Backup Files Not Yet Replicated**

The number of backup files that are not yet replicated to the target replication server.

**Archive Files Not Yet Replicated**

The number of archive files that are not yet replicated to the target replication server.

**Space Managed Files Not Yet Replicated**

The number of space-managed files that are not yet replicated to the target replication server.

**Backup Files To Delete**

The number of backup files to be deleted on the target replication server.

**Archive Files To Delete**

The number of archive files to be deleted on the target replication server.

**Space Managed Files To Delete**

The number of space-managed files to be deleted on the target replication server.

**Backup Files Deleted**

The number of backup files that are deleted on the target replication server.

**Archive Files Deleted**

The number of archive files that are deleted on the target replication server.

**Space Managed Files Deleted**

The number of space-managed files that are deleted on the target replication server.

**Backup Files Not Deleted Due To Errors**

The number of backup files that were not deleted from the target replication server because of errors.

**Archive Files Not Deleted Due To Errors**

The number of archive files that were not deleted from the target replication server because of errors.

**Space Managed Files Not Deleted Due To Errors**

The number of space-managed files that were not deleted from the target replication server because of errors.

**Backup Files To Update**

The number of backup files to update on the target replication server. If the metadata of a file is changed, the changed fields are sent to the target replication server.

**Archive Files To Update**

The number of archive files to update on the target replication server. If the metadata of a file is changed, the changed fields are sent to the target replication server.

**Space Managed Files To Update**

The number of space-managed files to update on the target replication server. If the metadata of a file is changed, the changed fields are sent to the target replication server.

**Backup Files Updated**

The number of backup files that are updated on the target replication server.

**Archive Files Updated**

The number of archive files that are updated on the target replication server.

**Space Managed Files Updated**

The number of space-managed files that are updated on the target replication server.

**Backup Files Not Updated Due To Errors**

The number of backup files that were not updated on the target replication server because of errors.

**Archive Files Not Updated Due To Errors**

The number of archive files that were not updated on the target replication server because of errors.

**Space Managed Files Not Updated Due To Errors**

The number of space-managed files that were not updated on the target replication server because of errors.

**Backup Bytes To Replicate (MB)**

The number of backup bytes to replicate to the target replication server.

**Archive Bytes To Replicate (MB)**

The number of archive bytes to replicate to the target replication server.

**Space Managed Bytes To Replicate (MB)**

The number of space-managed bytes to replicate to the target replication server.

**Backup Bytes Replicated (MB)**

The number of backup bytes that are replicated to the target replication server.

If a file was stored in a deduplicated storage pool, the number of bytes in the stored file might be less than the number of bytes in the original file. This field represents the number of physical bytes in the original file.

**Archive Bytes Replicated (MB)**

The number of archive bytes that are replicated to the target replication server.

If a file was stored in a deduplicated storage pool, the number of bytes in the stored file might be less than the number of bytes in the original file. This field represents the number of physical bytes in the original file.

**Space Managed Bytes Replicated (MB)**

The number of space-managed bytes that are replicated to the target replication server.

If a file was stored in a deduplicated storage pool, the number of bytes in the stored file might be less than the number of bytes in the original file. This field represents the number of physical bytes in the original file.

**Backup Bytes Transferred (MB)**

The number of backup bytes that were sent to the target replication server.

The value in this field represents the actual number of file bytes sent to the target replication server. This value is calculated by subtracting the number of bytes not sent because of deduplication from the number of bytes to replicate.

**Archive Bytes Transferred (MB)**

The number of archive bytes that were sent to the target replication server.

The value in this field represents the actual number of file bytes sent to the target replication server. This value is calculated by subtracting the number of bytes not sent because of deduplication from the number of bytes to replicate.

**Space Managed Bytes Transferred (MB)**

The number of space-managed bytes that were sent to the target replication server.

The value in this field represents the actual number of file bytes sent to the target replication server. This value is calculated by subtracting the number of bytes not sent because of deduplication from the number of bytes to replicate.

**Backup Bytes Not Replicated Due to Errors (MB)**

The number of backup bytes that were not replicated to the target replication server because of errors.

**Archive Bytes Not Replicated Due to Errors (MB)**

The number of archive bytes that were not replicated to the target replication server because of errors.

**Space Managed Bytes Not Replicated Due to Errors (MB)**

The number of space-managed bytes that were not replicated to the target replication server because of errors.

**Backup Bytes Not Yet Replicated (MB)**

The number of backup bytes not yet replicated to the target replication server.

**Archive Bytes Not Yet Replicated (MB)**

The number of archive bytes not yet replicated to the target replication server.

**Space Managed Bytes Not Yet Replicated (MB)**

The number of space-managed bytes not yet replicated to the target replication server.

**Total Files Needing No Action**

The total number of files in the file space that did not need to be replicated, updated, or deleted.

**Total Files To Replicate**

The total number of files to replicate to the target replication server.

**Total Files Replicated**

The total number of files that are replicated to the target replication server.

**Total Files Not Replicated Due To Errors**

The total number of files that were not replicated because of errors.

**Total files Not Yet Replicated**

The total number of files that are not yet replicated to the target replication server.

**Total Files To Delete**

The total number of files that were deleted on the target replication server.

**Total Files Deleted**

The total number of files that are deleted on the target replication server.

**Total Files Not Deleted Due to Errors**

The total number of backup, archive, and space-managed files that were not deleted on the target replication server because of errors.

**Total Files To Update**

The total number of files to be updated on the target replication server. When the metadata of a file is changed, the changed fields are sent to the target replication server.

**Total Files Updated**

The total number of files that are updated on the target replication server.

**Total Files Not Updated Due to Errors**

The total number of backup, archive, and space-managed files that were not updated on the target replication server because of errors.

**Total Bytes To Replicate (MB)**

The total number of bytes to replicate to the target replication server.

**Total Bytes Replicated (MB)**

The total number of bytes that are replicated to the target server.

If a file was stored in a deduplicated storage pool, the number of bytes in the stored file might be less than the number of bytes in the original file. This field represents the number of physical bytes in the original file.

**Total Bytes Transferred (MB)**

The total number of bytes that were transferred to the target replication server.

For files stored in a deduplicated storage pool, the value in this field includes the number of bytes in the original file before duplicate extents were removed. If duplicate extents were already on the target replication server, the number of bytes in the original file is more than the number of bytes transferred.

**Total Bytes Not Replicated Due to Errors (MB)**

The total number of bytes that were skipped because the source replication server was unable to transfer them to the target replication server.

**Total Bytes Not Yet Replicated (MB)**

The total number of bytes not yet transferred to the target replication server.

**Estimated Percentage Complete**

The estimated completion percentage that is based on the number of bytes.

**Estimated Time Remaining**

The estimated time that remains before the node replication process is complete.

**Estimated Time Of Completion**

The estimated time when the node replication process ends.

*Table 256. Commands related to QUERY REPLICATION*

Command	Description
CANCEL REPLICATION	Cancels node replication processes.

Table 256. Commands related to QUERY REPLICATION (continued)

<b>Command</b>	<b>Description</b>
QUERY ACTLOG	Displays messages from the server activity log.
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 PROCESS	Displays information about background processes.
QUERY REPLNODE	Displays information about the replication status of a client node.
QUERY REPLRULE	Displays information about node replication rules.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
SET REPRETENTION	Specifies the retention period for replication history records.



## QUERY REPLNODE (Display information about replication status for a client node)

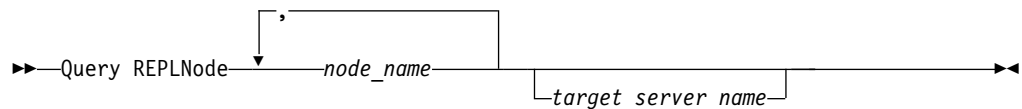
Use this command to display the number of files that are stored for each replicated file space. Information is displayed about file spaces for every client node that is configured for replication.

A client node is configured for replication if it is enabled or disabled.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

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

Specifies the client node that owns the files about which you want information. You can specify one or more names. If you specify multiple names, separate the names with commas. Do not use intervening spaces. You can use wildcard characters to specify multiple names.

Information about client nodes that match the file criteria, but that are not configured for replication, is not displayed.

#### *target\_server\_name*

Specifies the name of the replication server to query for replication information. This parameter is optional. If you do not specify a value for this parameter, the server that is the default target for replicated data is queried.

As the value for this parameter, you can also specify a server that was formerly a target for replicated data.

The client nodes that are defined to a replication server can be the source or the target of replicated data. To determine whether a particular client node is sending or receiving data, issue the **QUERY NODE** command. Look for the value *Send* or *Receive* in the Replication Mode field of the output.

To display the name of the active target replication server, issue the **QUERY STATUS** command, and look for the name in the Target Replication Server field.

### Example: List client node files on a source and a target replication server

The name of the client node is NODE1.

```
query replnode *
```

Node Name	Type	Filespace Name	FSID	Files on Server	Replication Server (1)	Files on Server (1)
NODE1	SpMg	/hmsmfs	1	1		
NODE1	Bkup	/1space2	2	27		
NODE1	Arch	/1space2	2	22	TGTSRV	22
NODE1	Bkup	/1space	3	18,096		
NODE1	Arch	/1space	3	61,150	TGTSRV	61,150
NODE2						

The number of files that are displayed for the replication servers might be different for the following reasons:

- The output of the **QUERY REPLNODE** command displays the number of files obtained from the occupancy table. The occupancy table contains only files that have a length greater than zero. Files that have a length of 0 and have been replicated are not reflected in this output.
- If only active data is replicated to the target server, the number of files that are displayed for the source server will be larger than the number of files that are displayed on the target server. The reason for the difference is that the source replication server has both active and inactive data, and the target server has only active data.
- A client node might have data that was exported from the source replication server and imported to the target replication server. If that data was synchronized and if the client node also stored data to the target replication server, then the number of files on the target replication server will be greater than the number of files stored as a result of export-and-import operations and replication.
- When you replicate node data from a source server prior to version 7.1, to a target server at version 7.1 or later, files that are larger than 10 GB are split in to smaller files if the **SPLITLARGEOBJECTS** parameter for the node definition is set to Yes. Each of these split files are counted on the target server.

### Field descriptions

**Node Name**

The name of the client node that owns the files.

**Type** The type of data. If this field is blank, the client node is configured for replication, but it does not have data on the replication server. In the example output, NODE2 is configured for replication, but it does not have backup, archive, or space-managed data.

The following values are possible:

- Arch** Archive data
- Bkup** Backup data
- SpMg** Data that was migrated by IBM Tivoli Storage Manager for Space Management clients

**Filespace Name**

The name of the file space that belongs to the node.

If this field is blank, the client node is configured for replication, but it does not have data on the replication server.

File space names can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative

command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

**FSID** The file space identifier for the file space. The server assigns a unique FSID when a file space is initially stored on the server. If this field is blank, the client node is configured for replication, but it does not have data on the replication server.

#### **Files on Server**

The number of backup, archive, and space-managed files on the server on which this command is issued. If this field is blank, the client node is configured for replication, but it does not have data on the replication server.

#### **Replication Server (1)**

The name of the replication server that is being queried for information. If this field is blank, one or more of the following conditions might exist:

- The file space of the node on the replication server where the command was issued does not have any data.
- The client node is not defined on replication server (1).
- The client node is defined on replication server (1), but the node is not configured for replication.
- The corresponding file space on replication server (1) does not have data or the file space is not defined.

#### **Files on Server (1)**

The number of files for the data type that are stored on the target replication server. This field can be blank. If it is, one or more of the following conditions might exist:

- Replication server (1) does not have any data.
- The client node is not defined on replication server (1).
- The client node is defined on replication server (1), but the node is not configured for replication.
- The corresponding file space on replication server (1) does not have data or the file space is not defined.

## **Related commands**

*Table 257. Commands related to QUERY REPLNODE*

<b>Command</b>	<b>Description</b>
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.

*Table 257. Commands related to QUERY REPLNODE (continued)*

<b>Command</b>	<b>Description</b>
QUERY REPLICATION	Displays information about node replication processes.
QUERY REPLRULE	Displays information about node replication rules.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
UPDATE REPLRULE	Enables or disables replication rules.

## QUERY REPLRULE (Query replication rules)

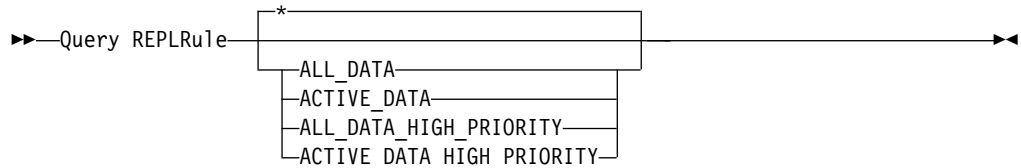
Use this command to display information about replication rules.

Issue this command on the server that acts as a source for replicated data.

### Privilege class

Any administrator can issue this command.

### Syntax



### Privilege class

Any administrator can issue this command.

### Parameters

*rule\_name*

Specifies the name of the replication rule that you want to display information about. This parameter is optional. You can use wildcard characters to specify one or more rules. If you do not specify this parameter, information about all rules is displayed in the query output. You can specify the following values:

#### ALL\_DATA

Displays information about the ALL\_DATA replication rule. This rule replicates backup, archive, or space-managed data. The data is replicated with a normal priority.

#### ACTIVE\_DATA

Displays information about ACTIVE\_DATA replication rule. This rule replicates only active backup data. The data is replicated with a normal priority. This rule is not valid for archive or space-managed data.

**Attention:** If you specify ACTIVE\_DATA and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the **FORCERECONCILE=YES** parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

### **ALL\_DATA\_HIGH\_PRIORITY**

Displays information about the ALL\_DATA\_HIGH\_PRIORITY rule. This rule replicates backup, archive, or space-managed data. The data is replicated with a normal priority. In a replication process, high-priority data is replicated before normal-priority data.

### **ACTIVE\_DATA\_HIGH\_PRIORITY**

Displays information about the ACTIVE\_DATA\_HIGH\_PRIORITY rule.

This rule is the same as the ACTIVE\_DATA replication rule except data is replicated with a high priority.

## **Example: Display information about a server replication rule**

The name of the rule is ALL\_DATA\_HIGH\_PRIORITY

```
query replrule all_data_high_priority
```

```
Replication Rule Name: ALL_DATA_HIGH_PRIORITY
Target Replication Server:
Active Only: No
Enabled: Yes
```

## **Field descriptions**

### **Replication Rule Name**

Specifies the name of the rule that was queried.

### **Target Replication Server**

Specifies the name of the target replication server.

### **Active Only**

Specifies whether the rule applies only to active backup data. The following values are possible:

- Yes** Specifies that only active backup data is replicated for file spaces to which this rule is assigned.
- No** Specifies that all backup data is replicated for file spaces to which this rule is assigned.

### **Enabled**

Specifies whether the rule is enabled or disabled. The following values are possible:

- Yes** Specifies that the rule is enabled for replication. Data in file spaces to which the rule is assigned is replicated.
- No** Specifies that the rule is not enabled for replication. Data in file spaces to which the rule is assigned is not replicated.

## **Related commands**

*Table 258. Commands related to QUERY REPLRULE*

<b>Command</b>	<b>Description</b>
QUERY REPLICATION	Displays information about node replication processes.
QUERY REPLNODE	Displays information about the replication status of a client node.
UPDATE REPLRULE	Enables or disables replication rules.

## QUERY REPLSERVER (Query a replication server)

Use this command to view information about all replication servers that are known server. The output from this command includes server information for the server from which the command was issued. The command indicates whether a replication server definition is deleted as a result of a **REMOVE REPLSERVER** command.

### Privilege class

Any administrator can issue this command.

### Syntax

►►—Query REPLServer—►►

### Example: Display summary statistics about the replicating server

Display information about the replicating server. Issue the command from either the source or the target replication server:

```
query replserver
```

```
Replication Globally Unique ID: 4d.83.fc.30.67.c1.11.e1.b8.40.f0.de.f1.5e.f1.89
  Server Name: Server1
  Last Replication:
  Heartbeat:
  Failover High Level Address: server1.example.com
  Failover TCP Port Number: 1500
  Failover SSL Port Number: 1542
  Deletion in Progress: No
  Dissimilar Policies:

Replication Globally Unique ID: 91.0f.ef.90.5c.cc.11.e1.ae.34.08.00.27.00.58.dc
  Server Name: DRServer1
  Last Replication: 06/30/2012 08:16:30 PM
  Heartbeat: 07/09/2012 22:15:22 PM
  Fail over High Level Address: drserver1.example.com
  Failover TCP Port Number: 1500
  Failover SSL Port Number: 1542
  Deletion in Progress: No
  Dissimilar Policies: On

Replication Globally Unique ID: 90.4f.53.b0.8e.cb.11.e3.a8.2f.00.14.5e.55.b3.67
  Server Name: DRSERVER2
  Last Replication: 04/01/14 12:38:28
  Heartbeat: 05/29/14 11:15:44
  Failover High Level Address: drserver2.example.com
  Failover TCP Port Number: 1500
  Failover SSL Port Number:
  Deletion in Progress: No
  Dissimilar Policies: Off
```

### Field descriptions

#### Replication Globally Unique ID

The unique identifier for the Tivoli Storage Manager server. The values for the Replication Globally Unique ID are created when a server is first used in a replication process.

**Tip:** The ID listed in the Replication Globally Unique ID field is not the same value as the value for the ID listed in the Machine Globally Unique ID field that is shown in the **QUERY STATUS** command.

**Server Name**

The name of the replication server.

**Last Replication**

The date of the last replication process that used the server.

**Heartbeat**

The last time that the server completed a successful test communication session.

**Failover TCP Port Number**

The active Transmission Control Protocol (TCP) client port on the replication server that is used for client connections. If the client is configured for TCP, the port is used to connect to the failover server.

**Failover SSL Port Number**

The active Secure Sockets Layer (SSL) port on the replication server that is used for client connections. If the client is configured for SSL, the port is used to connect to the failover server.

**Failover High Level Address**

The high-level address that the client uses to connect to the replication server during failover.

**Deletion in Progress**

Specifies whether a **REMOVE REPLSERVER** command was issued for this replication server and is still in progress. The following values are possible:

**Yes** The deletion of the replication server is in progress.

**No** The deletion of the replication server is not in progress.

**Dissimilar Policies**

Specifies whether the policies that are defined on the target replication server are enabled. The following values are possible:

**On** The policies on the target replication server manage replicated client-node data.

**Off** The policies on the source replication server manage replicated client-node data.

**Related commands**

*Table 259. Commands related to QUERY REPLSERVER*

Command	Description
"REMOVE REPLNODE (Remove a client node from replication)" on page 1012	Removes a node from replication.
"REMOVE REPLSERVER (Remove a replication server)" on page 1014	Removes a server from replication.



## QUERY REQUEST (Query one or more pending mount requests)

Use the **QUERY REQUEST** command to show information about one or more pending mount requests. The server makes requests for the administrator to complete an action, like inserting a tape volume in a library after a **CHECKIN LIBVOL** is issued.

### Privilege class

Any administrator can issue this command.

### Syntax

```
►► Query REQuest request_number ►►
```

### 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 about all pending mount requests after a **CHECKIN LIBVOL** is issued.

```
query request
```

### Output for a manual Library

### Output for an automated Library

```
ANR8352I Requests outstanding:  
ANR8306I 001: Insert LTO volume 133540L5 R/W into the slot with element  
number 31 of library LTOLIB within 60 minutes; issue 'REPLY'  
along with the request ID when ready.
```

### Related commands

Table 260. Related commands for **QUERY REQUEST**

Command	Description
CANCEL REQUEST	Cancels pending volume mount requests.
REPLY	Allows a request to continue processing.

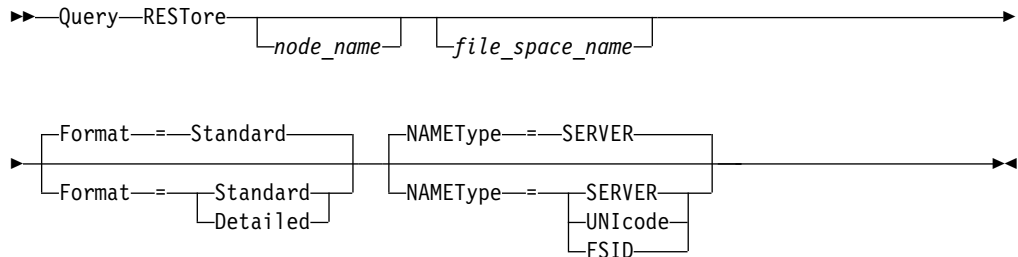
## QUERY RESTORE (Query restartable restore sessions)

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/TESTDIR\
```

## 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 can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed

up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

**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 261. Commands related to QUERY RESTORE*

<b>Command</b>	<b>Description</b>
CANCEL RESTORE	Cancel 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 specifies the device class REMOTE. The output of this command is the entire contents of the 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.

```
query rpfcontent branch1.19980319.061000 nodename=polaris
```

## Related commands

*Table 262. 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.

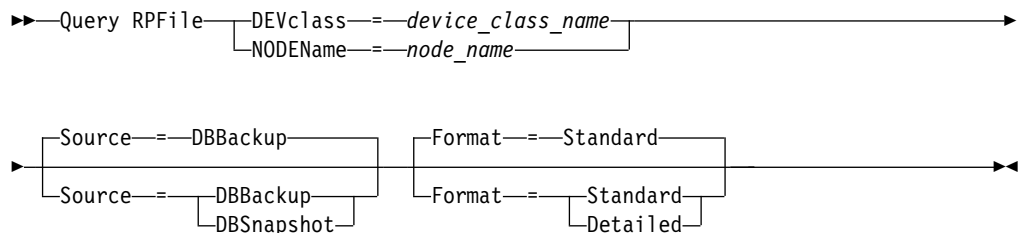
## QUERY RPFIL (Query recovery plan file information stored on a target server)

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 rpf file devclass=* format=detailed
```

```
Recovery Plan File Name: ALASKA.20000406.170423
      Node Name: BRANCH1
      Device Class Name: REMOTE
Recovery Plan File Type: RPF FILE
      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: RPF SNAPSHOT
      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 rpf file 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:

#### RPF FILE

The plan assumes full plus incremental database backups.

#### RPF SNAPSHOT

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 263. Commands related to QUERY RPFFILE*

<b>Command</b>	<b>Description</b>
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 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.

- “QUERY SCHEDULE (Query an administrative schedule)” on page 897
- “QUERY SCHEDULE (Query client schedules)” on page 893

*Table 264. Commands related to QUERY SCHEDULE*

<b>Command</b>	<b>Description</b>
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.

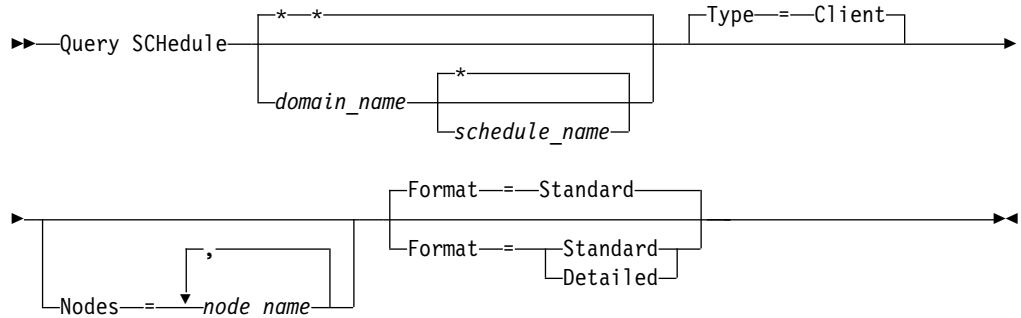
## QUERY SCHEDULE (Query client schedules)

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
EMPLOYEE_RECORDS	WEEKLY_BACKUP	Inc Bk	2004.06.04 17.04.20	1 H	1 D	Any
EMPLOYEE_RECORDS	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 895 for field descriptions.

```
query schedule * type=client format=detailed
```

```

Policy Domain Name: ADMIN_RECORDS
Schedule Name: ADMIN_BACKUP
Description:
  Action: Backup
  Subaction: vApp
  Options:
  Objects:
  Priority: 5
Start Date/Time: 04/06/2013 17.04.20
Duration: 1 Hour(s)
Maximum Run Time (Minutes): 0
Schedule Style: Classic
Period: 1 Day(s)
Day of Week: Any
Month:
Day of Month:
Week of Month:
Expiration:
Last Update by (administrator): $$CONFIG MANAGERS$$
Last Update Date/Time: 04/06/2013 17.51.49
Managing profile: ADMIN_INFO

Policy Domain Name: EMPLOYEE_RECORDS
Schedule Name: EMPLOYEE_BACKUP
Description:
  Action: Incremental
  Subaction:
  Options:
  Objects:
  Priority: 5
Start Date/Time: 2004.06.04 17.04.33
Duration: 1 Hour(s)
Maximum Run Time (Minutes): 0
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): $$CONFIG MANAGERS$$
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.

### Subaction

Specifies that the type of operation identified by the **ACTION** parameter is to be scheduled. See the DEFINE SCHEDULE command for a listing of subactions.

### 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.
- Maximum Run Time (Minutes)**  
Specifies the number of minutes during which all client sessions that are started by the scheduled operation should be completed. If sessions are still running after the maximum run time, the server issues a warning message, but the sessions continue to run.
- 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.

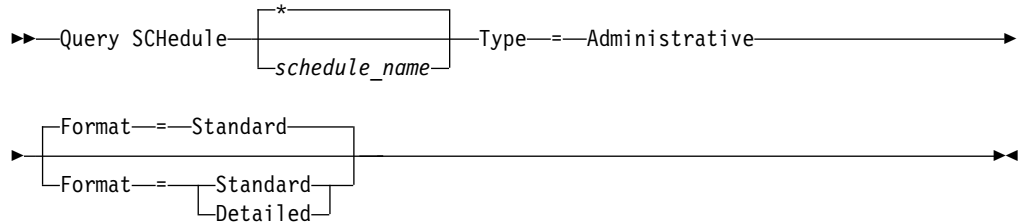
## QUERY SCHEDULE (Query an administrative schedule)

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 898 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)
Maximum Run Time (Minutes): 0
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)
Maximum Run Time (Minutes): 0
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.

### Maximum Run Time (Minutes)

Specifies the number of minutes during which server processes that are started by the scheduled commands must be completed. If processes are still running after the maximum run time, the central scheduler cancels the processes.

### Tips:



- This parameter does not apply to some processes, such as duplicate-identification processes, which can continue to run after the maximum run time.
- Another cancel time might be associated with some commands. For example, the **MIGRATE STGPOOL** command can include a parameter that specifies the length of time that the storage pool migration runs before the migration is automatically canceled. If you schedule a command for which a cancel time is defined, and you also define a maximum run time for the schedule, the processes are canceled at whichever cancel time is reached first.

**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.

**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.

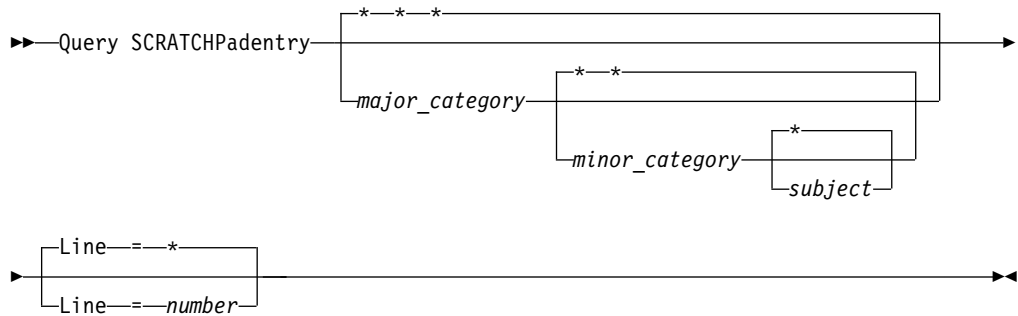
## QUERY SCRATCHPADENTRY (Query a scratch pad entry)

Use this command to display data that is contained in the scratch pad.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *major\_category*

Specifies the major category to be queried. This parameter is case sensitive. You can query all major categories by omitting this parameter.

#### *minor\_category*

Specifies the minor category to be queried. This parameter is case sensitive. You can query all minor categories in the major category by omitting this parameter.

#### *subject*

Specifies the subject to be queried. This parameter is case sensitive. You can query all subjects in the minor category by omitting this parameter.

#### **Line**

Specifies the number of the line to be queried. For *number*, enter an integer in the range 1 - 1000. You can query all lines of data in the subject by omitting this parameter.

### Example: Query scratch pad entries

Query a database that stores information about the location of all administrators.

```
query scratchpadentry admin_info location
```

```

Scratchpad major category: admin_info
  Scratchpad minor category: Location
    Scratchpad subject: codjo
    Scratchpad line number: 1
    Scratchpad data: Toronto 5A24
    Date/time of creation: 2013-09-10, 10:15:50
    Last Update Date/Time: 2013-09-10, 10:15:50
Last Update by (administrator): CODJO

Scratchpad major category: admin_info
  Scratchpad minor category: location
    Scratchpad subject: jane
    Scratchpad line number: 1
    Scratchpad data: Raleigh GF85
    Date/time of creation: 2013-09-09, 14:29:40
    Last Update Date/Time: 2013-09-09, 14:29:40
Last Update by (administrator): JANE_W

Scratchpad major category: admin_info
  Scratchpad minor category: location
    Scratchpad subject: jane
    Scratchpad line number: 2
    Scratchpad data: Out of the office from 1-15 Nov.
    Date/time of creation: 2013-09-09, 14:30:05
    Last Update Date/Time: 2013-10-31, 16:55:52
Last Update by (administrator): JANE_W

Scratchpad major category: admin_info
  Scratchpad minor category: location
    Scratchpad subject: montse
    Scratchpad line number: 1
    Scratchpad data: Barcelona B19
    Date/time of creation: 2013-09-10, 04:34:37
    Last Update Date/Time: 2013-09-10, 04:34:37
Last Update by (administrator): MONTSERRAT

```

## Field descriptions

### Scratchpad data

The data that is stored in the scratch pad entry.

### Date/time of creation

The date and time at which the scratch pad entry was created.

### Last Update Date/Time

The date and time at which the scratch pad entry was last updated.

### Last Update by (administrator)

The administrator who last updated the scratch pad entry.

## Related commands

Table 265. Commands related to **QUERY SCRATCHPADENTRY**

Command	Description
DEFINE SCRATCHPADENTRY	Creates a line of data in the scratch pad.
DELETE SCRATCHPADENTRY	Deletes a line of data from the scratch pad.
SET SCRATCHPADRETENTION	Specifies the amount of time for which scratch pad entries are retained.
UPDATE SCRATCHPADENTRY	Updates data on a line in the scratch pad.

## QUERY SCRIPT (Query Tivoli Storage Manager scripts)

Use this command to display information about scripts.

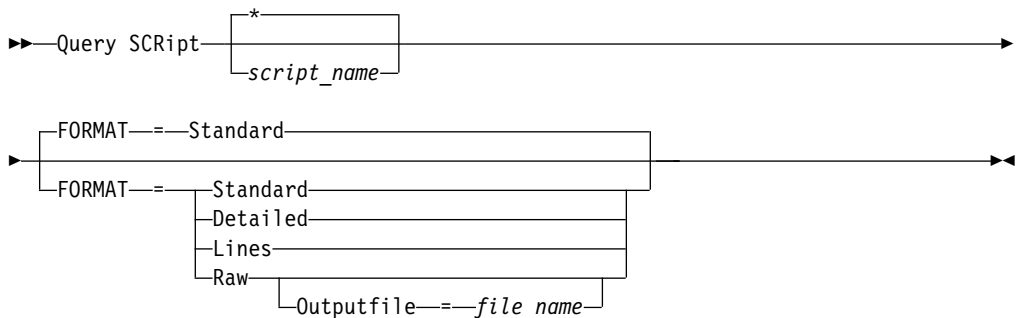
You can use this command with the **DEFINE SCRIPT** command to create a new script by using the contents from another script.

### Privilege class

The privilege class that is 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 that is used to process this command and the amount of information that is 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.

##### **Detailed**

Specifies that detailed information about the script is displayed. This information includes the commands in the script and their line numbers, the date of the last update and the administrator that completed the updates.

**Lines**

Specifies that the script name, the line number of the commands, comment lines, and the commands in the script are displayed.

**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 by 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 that you specify must be on the server that is 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 of 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: Display detailed script information**

Display detailed information about scripts. See "Field descriptions" on page 904 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 that is displayed in the previous field.

**Last Update by (administrator)**

The name of the administrator that 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 266. Commands related to QUERY SCRIPT*

Command	Description
COPY SCRIPT	Creates a copy of a script.
DEFINE SCRIPT	Defines a script to the 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.

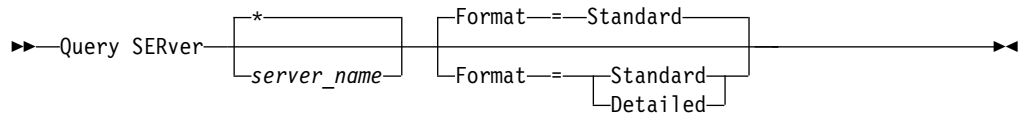
## QUERY SERVER (Query a server)

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 906 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 about a specific server

From a managed server, display detailed information about SERVER\_A. See “Field descriptions” on page 906 for field descriptions.

```
query server server_a format=detailed
```

```

Server Name: SERVER_A
Comm. Method: TCPIP
Transfer Method: TCPIP
High-level Address: 9.115.4.15
Low-level Address: 1500
Description:
Allow Replacement: No
Node Name:
Last Access Date/Time: 07/09/2013 09:00:00
Days Since Last Access: <1
Compression: Client's choice
Archive Delete Allowed?: No
URL:
Registration Date/Time: 07/08/2013 09:15:09
Registering Administrator: $$CONFIG_MANAGER$$
Bytes Received Last Session: 362
Bytes Sent Last Session: 507
Duration of Last Session: 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: Yes
Server Password Set Date/Time: 07/08/2013 09:15:09
Days Since Server Password Set: 1
Invalid Sign-on Count for Server: 0
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
Version: 7
Release: 1
Level: 0.0
Role(s): Replication
SSL: No

```

## Field descriptions

### Server Name

The name of the server.

### Comm. Method

The communication method that is used to connect to the server.

### Transfer Method

The method that is used for server-to-server data transfer.

### 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

Specifies 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.

**Compression**

The type of compression that is completed by Tivoli Storage Manager on client files.

**Archive Delete Allowed?**

Specifies 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**

The length of the last client node session, in seconds.

**Pct. Idle Wait Last Session**

The percentage of the total session time during which the client did not complete 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**

The number of days an object remains on the target server after it is marked for deletion.

**Managing Profile**

The profile from which the managed server got the definition of this server.

**Server Password Set**

Specifies whether the password for the server is set.

**Server Password Set Date/Time**

Specifies when the password for the server is 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 can accept.

**Virtual Volume Password Set**

Specifies whether the password used to log on to the target server is set.

**Virtual Volume Password Set Date/Time**

Specifies when the password for virtual volume support is 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 the data validation function enabled.

**Version**

The software version of the Tivoli Storage Manager server.

**Release**

The software release of the Tivoli Storage Manager server.

**Level** The software level of the Tivoli Storage Manager server.

**Role(s)**

The role of the server. For example, one of the roles that the server is used for is replication.

**SSL** Specifies whether Secure Sockets Layer (SSL) communication is used.

**Related commands**

*Table 267. 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.
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
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 node 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.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
SET REPLSERVER	Specifies a target replication server.
UPDATE DEVCLASS	Changes the attributes of a device class.

*Table 267. Commands related to QUERY SERVER (continued)*

<b>Command</b>	<b>Description</b>
UPDATE NODE	Changes the attributes that are associated with a client node.
UPDATE SERVER	Updates information about a server.

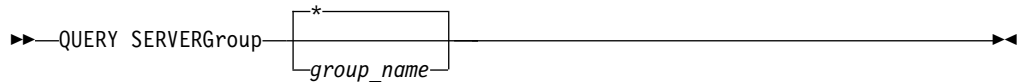
## QUERY SERVERGROUP (Query a server group)

Use this command to display information about server groups and group members.

### Privilege class

Any administrator can issue this command.

### 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	Members	Description	Managing Profile
ADMIN_GROUP	SERVER_A SERVER_B SERVER_C SERVER_D	Headquarters	ADMIN_INFO

### Field descriptions

#### Server Group

The name of the server group.

#### Members

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 268. 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 268. Commands related to QUERY SERVERGROUP (continued)*

<b>Command</b>	<b>Description</b>
RENAME SERVERGROUP	Renames a server group.
UPDATE SERVERGROUP	Updates a server group.

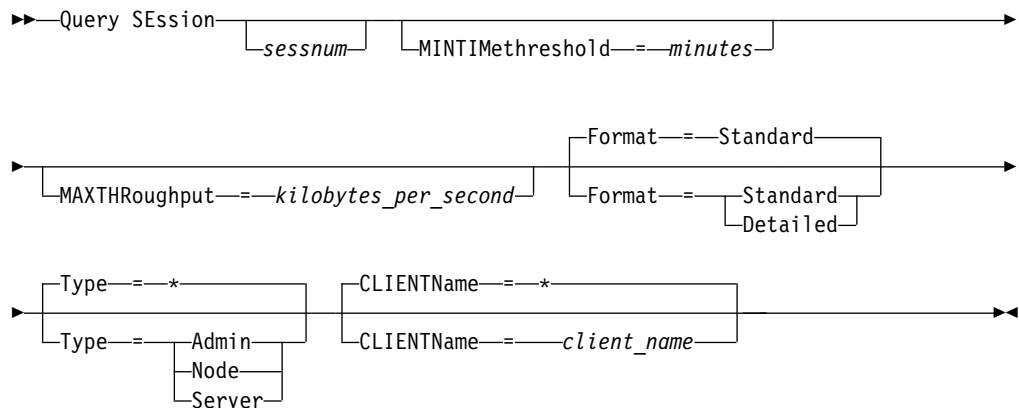
## QUERY SESSION (Query client sessions)

Use this command to display information about administrative, node, and server 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 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. The following values are possible:

##### **Standard**

Specifies that partial information displays for the session.

##### **Detailed**

Specifies that complete information displays for the session.

**Type** Specifies the type of sessions to include in the query results. If you do not specify a value for this parameter, all types of sessions are queried. This parameter is optional. You can specify one of the following values:

**Admin**

Specifies that administrative sessions are displayed.

**Node** Specifies that node sessions are displayed.

**Server** Specifies that server sessions are displayed.

**CLIENTName**

Specifies the name of an administrator, client node, or server to be queried. You can specify one or more names. You can also specify node groups and proxy nodes. If you specify multiple names, separate the names with commas; use no intervening spaces. You can use wildcard characters with node names but not with node group names. The parameter is optional.

During node replication, the client name on the target server is displayed as *node\_name (server\_name)*, where *node\_name* is the node whose data is being replicated, and *server\_name* is the name of the source server. You can specify either the node name or the server name in the **CLIENTName** parameter to display the replication sessions.

**Example: List active client node sessions**

Display information about all administrative and client node sessions that are communicating with the server. See “Field descriptions” for field descriptions.

query session

Sess Number	Comm. Method	Sess State	Wait Time	Bytes Sent	Bytes Recvd	Sess Type	Platform	Client Name
4	TCP/IP	Run	0 S	1.4 K	162	Admin	WinNT	ADMIN

**Example: Display detailed information about active client node sessions**

Display detailed information 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: 4
Comm. Method: Tcp/Ip
Sess State: Run
Wait Time: 0 S
Bytes Sent: 1.4 K
Bytes Recvd: 162
Sess Type: Admin
Platform: WinNT
Client Name: ADMIN
Media Access Status:
User Name:
Date/Time First Data Sent:
Proxy By Storage Agent:
Actions:
Failover Mode: No

```

**Field descriptions****Sess Number**

Specifies a unique session identification number that is assigned by the server.

**Comm. Method**

Specifies the method that is used by the client to communicate with the server.

**Sess State**

Specifies the current communications state of the server. The following states are possible:

**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 running 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 was already sent).

**SSLiW**

The session is waiting for Secure Sockets Layer (SSL) initialization to complete.

**Start** The session is starting (authentication is in progress).

**Wait Time**

Specifies the amount of time (seconds, minutes, or hours) the server is in the current state shown.

**Bytes Sent**

Specifies the number of bytes of data that is sent to the client node since the session was initiated.

**Bytes Recvd**

Specifies the number of bytes of data that is 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 that is associated with the client.

**Client Name**

Specifies the name of the client node or the administrator.

For node replication sessions, the client name is updated to *node\_name* (*server\_name*) on the target server after data transfer starts.

**Media Access Status**

Specifies the type of media wait state. 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 that is associated with the FILE or CENTERA device type. As a result, a volume in a storage pool with a device type of FILE or CENTERA 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 are performed during the session. An action is listed only once, even if the action occurs multiple times during a session. The following actions are possible:

**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 that are stored on the server are deleted.

**BkRebind**

One or more backup objects that are stored on the server are 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 are 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 are deleted.

**MigRebind**

One or more space-managed files that are stored on the server are bound to a different management class.

**MigRecall**

One or more space-managed files that are stored on the server are recalled.

**MigUpd**

The attributes for one or more space-managed files that are stored on the server are 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 is processed by the client node, which caused an authorization rule for access to the client node's data to be added.

**Failover Mode**

Specifies whether the client session was started in failover mode. The following values are possible:

- Force** The **FORCEFAILOVER** flag is specified on the client and the session is forced into failover mode.
- Yes** The client session was started in failover mode.
- No** The client session was not started in failover mode.

**Related commands**

*Table 269. Command related to QUERY SESSION*

Command	Description
CANCEL SESSION	Cancels active sessions with the server.

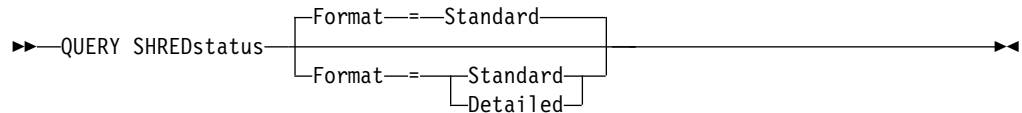
## QUERY SHREDSTATUS (Query shredding status )

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 Active	Objects 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 Active	Objects Awaiting Shred	Occupied Space (MB)	Data Left To Shred (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.

**Data Left to Shred (MB)**

The amount of data that still needs to be shredded.

**Related commands**

*Table 270. Commands related to QUERY SHREDSTATUS*

<b>Command</b>	<b>Description</b>
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 or directly to another server.
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.

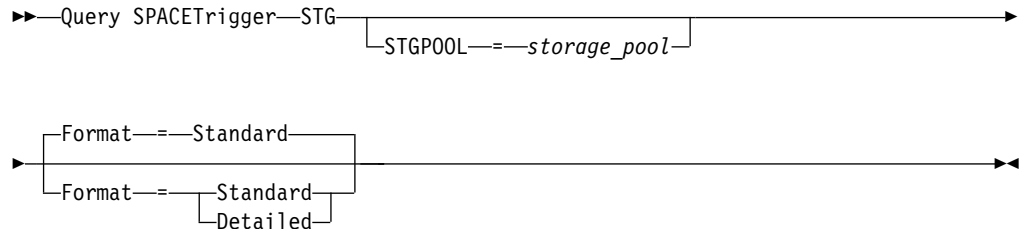
## QUERY SPACETRIGGER (Query the space triggers)

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: /opt/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 Expansion Percentage

The percentage of space by which the storage pool should be expanded.

**STGP00L Expansion prefix**

The prefix associated with the space trigger.

**STGP00L**

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 271. Commands related to QUERY SPACETRIGGER*

<b>Command</b>	<b>Description</b>
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 SSLKEYRINGPW (Query SSL key database file password)

Use this command to display the Secure Sockets Layer (SSL) key database file password. The key database file password is needed in order to update the key database file with new certificates or to designate a default certificate.

### Privilege class

You must have system privileges to issue this command.

### Syntax

►►—Query SSLKEYRINGPW—————►►

### Parameters

The **QUERY SSLKEYRINGPW** command contains no parameters.

### Example: Display the key database file password

Issue the command:

```
query sslkeyringpw
```

```
214qX{;8TP
```

### Related commands

Table 272. Commands related to **QUERY SSLKEYRINGPW**

Command	Description
DELETE KEYRING	Deletes password information in the certificate key database.
SET SSLKEYRINGPW	Sets or updates the key database file password.

## QUERY STATUS (Query system parameters)

Use the **QUERY STATUS** command to display information about system parameters.

Use this command for the following reasons:

- 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 that is 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 repeated attempts that are allowed after a failed attempt to run a scheduled command
- To display whether subfiles can be backed up to this server, as indicated by the **SET SUBFILE** command
- To display information about a target replication server
- To display licensing information

**Tip:** To display information about a target replication server, you must issue the command from the target replication server.

### 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 run from a configuration manager. For descriptions of displayed fields, see “Field descriptions” on page 924.

```
query status
```



```

Server Name: RATBERT_IVT
Server host name or IP address: ratbert
Server TCP/IP port number: 1510
Crossdefine: Off
Server Password Set: Yes
Server Installation Date/Time: 2015-07-08, 01:05:24
Server Restart Date/Time: 2015-11-10, 18:10:40
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
Inbound Sessions Disabled:
Outbound Sessions Disabled:
Accounting: Off
Activity Log Retention: 30 Day(s)
Activity Log Number of Records: 62429
Activity Log Size: 1 M
Activity Summary Retention Period: 30 Day(s)
License Audit Period: 30 Day(s)
Last License Audit: 2015-10-21, 20:51:17
Server License Compliance: Valid
Central Scheduler: Active
Maximum Sessions: 200
Maximum Scheduled Sessions: 100
Event Record Retention Period: 30 Day(s)
Client Action Duration: 5 Day(s)
Schedule Randomization Percentage: 25
Query Schedule Period: Client
Maximum Command Retries: Client
Retry Period: Client
Client-side Deduplication Verification Level: 0 %
Scheduling Modes: Any
Active Receivers: CONSOLE ACTLOG
Configuration manager?: Off
Refresh interval: 60
Last refresh date/time:
Context Messaging: On
Table of Contents (TOC) Load Retention: 120 Minute(s)
Machine Globally Unique ID: 00.00.00.01.8c.27.11.de.b5.a6.00.00.09.0b.7d.e3
Archive Retention Protection: Off
Database Directories: /tsmdb/ivt
Total Space of File System (MB): 101,376.00
Used Space on File System (MB): 77,251.12
Free Space Available (MB): 24,124.88
Encryption Strength: AES
Client CPU Information Refresh Interval: 180
Outbound Replication: Enabled
Target Replication Server: DUMPLING
Default Replication Rule for Archive: ALL_DATA
Default Replication Rule for Backup: ACTIVE_DATA
Default Replication Rule for Space Management: ALL_DATA
Replication Record Retention Period: 30 Day(s)
LDAP User:
LDAP Password Set: No
Default Authentication: Local
Failover High Level Address:
Scratchpad retention: 365 Day(s)
Replication Recovery of Damaged Files: On
SUR Occupancy (TB): 0
SUR Occupancy Date/Time: 2015-11-10, 05:39:33
Front-End Capacity (MB): 226,331
Front-End Client Count: 6
Front-End Capacity Date: 2015-11-13, 09:20:02
Product Offering: Tivoli Storage Manager

```

## 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.

### Crossdefine

Specifies whether another server that is running the **DEFINE SERVER** command automatically defines itself to this server. See the **SET CROSSDEFINE** command.

### Server Password Set

Specifies whether the password was 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.

### Inbound Sessions Disabled

Specifies the names of servers from which server-to-server communications are not allowed. To enable inbound server sessions, use the **ENABLE SESSIONS** command.

### Outbound Sessions Disabled

Specifies the names of servers to which server-to-server communications are not allowed. To enable outbound server sessions, use the **ENABLE SESSIONS** command.

### 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 Log Number of Records**

Specifies the number of records in the activity log.

**Activity Log Size**

Specifies the size of the activity log.

**Activity Summary Retention Period**

Specifies the number of days information is retained in the SQL activity summary table.

**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 that is defined with the **DEFINE CLIENTACTION** command.

**Schedule Randomization Percentage**

Specifies how much of the startup window is used for running 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 tries to run a scheduled command after a failed attempt. If the value in this field is Client, the client node determines the maximum number.

**Retry Period**

Specifies the number of minutes between failed attempts by the client scheduler to contact the server or to run a scheduled command. If the value in this field is Client, the client node determines the number of minutes.

**Client-side Deduplication Verification Level**

Specifies a percentage of extents to be verified by the Tivoli Storage Manager server. The extents are created during client-side data deduplication.

**Scheduling Modes**

Specifies the central scheduling modes that are supported by the server.

**Active Receivers**

Specifies the receivers for which event logging began.

**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.

**Table of Contents (TOC) Load Retention**

Specifies the approximate number of minutes that unreferenced TOC data is retained in the database.

**Machine Globally Unique ID**

The globally unique identifier (GUID) as of the last time that the server was started. This GUID identifies the host system to which the current server belongs.

**Archive Retention Protection**

Specifies whether archive data retention protection is activated or deactivated.

**Database Directories**

Specifies the locations of the database directories.

**Total Space of File System (MB)**

Specifies the total size of the file system.

**Used Space on File System (MB)**

Specifies the amount of space that is in use on the file system.

**Free Space Available (MB)**

Specifies the amount of space that is available.

**Encryption Strength**

Indicates data encryption strength: AES or DES.

**Client CPU Information Refresh Interval**

Specifies the number of days that elapse between client scans for CPU information that is used for PVU estimation.

**Outbound Replication**

Specifies whether replication processing is enabled or disabled. If outbound replication is disabled, new replication processes cannot start on the server.

### **Target Replication Server**

Specifies the name of the server that is the target for node replication operations. If a target replication server does not exist, this field is blank.

### **Default Replication Rule for Archive**

Specifies the server replication rule that applies to archive data. The following values are possible:

#### **ALL\_DATA**

Replicates archive data. The data is replicated with a normal priority.

#### **ALL\_DATA\_HIGH\_PRIORITY**

Replicates archive data. The data is replicated with a high priority.

#### **NONE**

Archive data is not replicated.

### **Default Replication Rule for Backup**

Specifies the server replication rule that applies to backup data. The following values are possible:

#### **ALL\_DATA**

Replicates active and inactive backup data. The data is replicated with a normal priority.

#### **ACTIVE\_DATA**

Replicates only active backup data. The data is replicated with a normal priority.

**Attention:** If you specify **ACTIVE\_DATA** and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the **FORCERECONCILE=YES** parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

#### **ALL\_DATA\_HIGH\_PRIORITY**

Replicates active and inactive backup data. The data is replicated with a high priority.

#### **ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the **ACTIVE\_DATA** replication rule except data is replicated with a high priority.

#### **NONE**

Backup data is not replicated.

### **Default Replication Rule for Space Management**

Specifies the server replication rule that applies to space-managed data. The following values are possible:

**ALL\_DATA**

Replicates space-managed data. The data is replicated with a normal priority.

**ALL\_DATA\_HIGH\_PRIORITY**

Replicates space-managed data. The data is replicated with a high priority.

**NONE**

Space-managed data is not replicated.

**Replication Record Retention Period**

Specifies the number of days that replication history records are retained in the database of the source replication server.

**LDAP User**

Specifies the user ID that is named in the **SET LDAPUSER** command. This user ID can issue administrative commands on the namespace that is reserved for Tivoli Storage Manager on the LDAP directory server.

**LDAP Password Set**

This output field shows if a password is defined for the user ID that is named in the **SET LDAPUSER** command. The values are YES and NO. If YES, the user ID that is named in the **SET LDAPUSER** command can issue administrative commands on the LDAP namespace that is reserved for Tivoli Storage Manager. If NO, issue the **SET LDAPPASSWORD** command to set the password for the user ID that is named in the **SET LDAPUSER** command.

**Default Authentication**

Specifies the default password authentication method: LOCAL or LDAP.

Authentication Target	Authentication Method
Tivoli Storage Manager server	LOCAL
LDAP directory server	LDAP

When you issue the **SET DEFAULTAUTHENTICATION** command, you define the resulting authentication method for all **REGISTER ADMIN** and **REGISTER NODE** commands. The default is LOCAL.

**Failover High Level Address**

Specifies the high-level address for the failover server that is used by the client. Client restore operations fail over to this high-level address when the interface that is used by the client is different from the interface that is used by replication.

**Scratchpad retention**

Specifies the number of days for which scratch pad entries are retained since they were last updated.

**Replication Recovery of Damaged Files**

Specifies whether node replication is enabled to recover damaged files from a target replication server. This is a system-side setting. If ON is specified, the node replication process can be configured to detect damaged files on a source replication server and replace them with undamaged files from a target replication server. If OFF is specified, damaged files are not recovered from a target replication server.

**SUR Occupancy (TB)**

If you have a Tivoli Storage Manager Suite for Unified Recovery (SUR) license, this field specifies the SUR occupancy on the server. The *SUR*

*occupancy* is the amount of space that is used to store data that is managed by the Tivoli Storage Manager products that are included in the SUR bundle.

**SUR Occupancy Date/Time**

Specifies the date and time when SUR occupancy data was last collected.

**Front-End Capacity (MB)**

Specifies the amount of primary data that is reported as being backed up by clients. Clients include applications, virtual machines, and systems. This value is used for the front-end licensing model.

**Front-End Client Count**

Specifies the number of clients that reported capacity usage based on the front-end licensing model.

**Front-End Capacity Date**

Specifies the date and time when front-end capacity data was last collected.

**Product Offering**

Specifies a product offering.

Value specified by the SET PRODUCTOFFERING command	Value shown in the QUERY STATUS command output
SSAM	System Storage Archive Manager
TSM	Tivoli Storage Manager
TSMEE	Tivoli Storage Manager Extended Edition
TSMSUR	Tivoli Storage Manager Suite for Unified Recovery
TSMSURENtry	Tivoli Storage Manager Suite for Unified Recovery Entry
TSMSURARCHive	Tivoli Storage Manager Suite for Unified Recovery - Archive
TSMSURPROtectier	Tivoli Storage Manager Suite for Unified Recovery - ProtecTier
TSMSURFRontend	Tivoli Storage Manager Suite for Unified Recovery - FrontEnd
TSMSURENTRYFRontend	Tivoli Storage Manager Suite for Unified Recovery Entry - FrontEnd
CLEAR	NULL

**Related commands**

*Table 273. Commands related to QUERY STATUS*

Command	Description
BEGIN EVENTLOGGING	Starts event logging to a specified receiver.
DISABLE REPLICATION	Prevents outbound replication processing on a server.
DISABLE SESSIONS	Prevents new sessions from accessing Tivoli Storage Manager but permits existing sessions to continue.
ENABLE REPLICATION	Allows outbound replication processing on a server.

Table 273. Commands related to QUERY STATUS (continued)

Command	Description
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 CPUINFOREFRESH	Specifies the number of days between client scans for workstation information used for PVU estimates.
SET CROSSDEFINE	Specifies whether to cross define servers.
SET DEDUPVERIFICATIONLEVEL	Specifies the percentage of extents verified by the server during client-side deduplication.
SET DEFAULTAUTHENTICATION	Specifies the default password authentication method for any REGISTER NODE or REGISTER ADMIN commands.
SET EVENTRETENTION	Specifies the number of days to retain records for scheduled operations.
SET LDAPPASSWORD	Sets the password for the LDAPUSER.
SET LDAPUSER	Sets the user who oversees the passwords and administrators on the LDAP directory server.
SET MAXCMDRETRIES	Specifies the maximum number of retries after a failed attempt to execute a scheduled command.
SET MAXSCHEDESESSIONS	Specifies the maximum number of client/server sessions available for processing scheduled work.
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.
SET PRODUCTOFFERING	Set the product offering licensed to your enterprise.
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.



Table 273. Commands related to QUERY STATUS (continued)

<b>Command</b>	<b>Description</b>
SET REPLRECOVERDAMAGED	Specifies whether node replication is enabled to recover damaged files from a target replication server.
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 STATUSTHRESHOLD (Query status monitoring thresholds)

Use this command to display information about status monitoring thresholds.

Status monitoring thresholds compare the defined conditions to the status monitoring server queries and inserts the results in the status monitoring table.

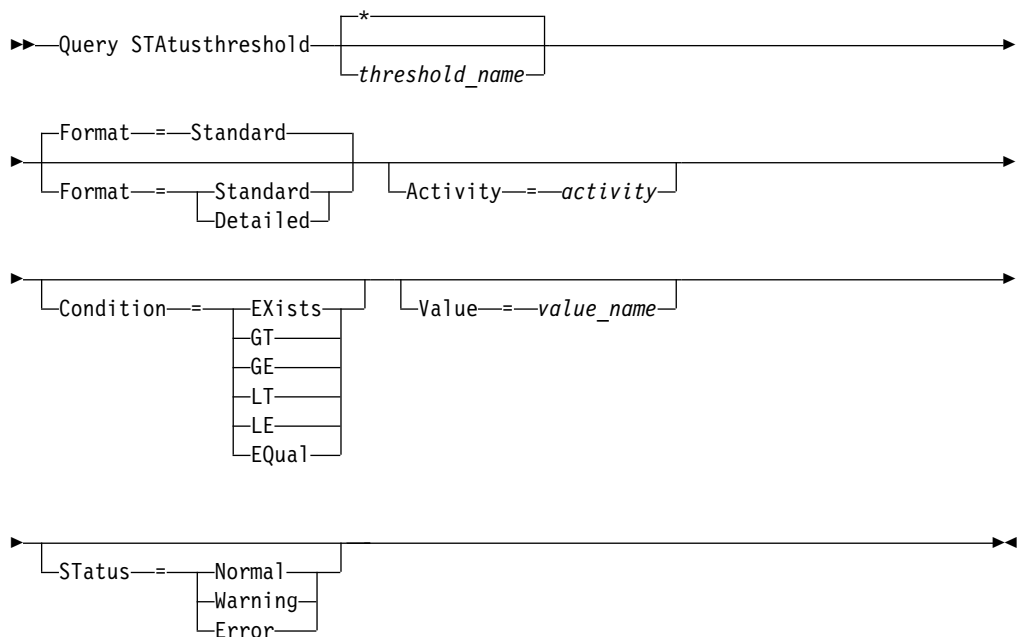
Multiple thresholds can be defined for an activity. For example, you can create a threshold that provides a warning status if storage pool capacity utilization is greater than 80%. You can then create another threshold that provides error status if storage pool capacity utilization is greater than 90%.

**Note:** If a threshold is already defined for an EXISTS condition, you cannot define another threshold with one of the other condition types.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

*threshold\_name*

Specifies the threshold name. The name cannot exceed 48 characters in length.

#### Format

Specifies how the information is displayed. The default value is `STANDARD`. Possible values are:

##### Standard

Specifies that partial information is displayed for the specified status thresholds.

**Detailed**

Specifies that complete information is displayed for the specified status thresholds.

*activity*

Specifies the activity for which you want to display status indicators. If you do not specify a value, information is displayed for all activities. For a list of activities, see the **DEFINE STATUSTHRESHOLD** command.

**Condition**

Restricts the output to only those matching the specified value. Possible values are:

**EXists**

Displays status thresholds where the condition equals EXISTS.

**GT** Displays status thresholds where the condition equals GT.

**GE** Displays status thresholds where the condition equals GE.

**LT** Displays status thresholds where the condition equals LT.

**LE** Displays status thresholds where the condition equals LE.

**EQual**

Displays status thresholds where the condition equals EQUAL.

**Value**

Displays thresholds that have the specified value. If you do not specify a value, information is displayed for all values. You can specify an integer from 0 to 9223372036854775807.

**Status**

Displays status thresholds that have the specified status value. If you do not specify a value, information is displayed for all values. Possible values are:

**Normal**

Displays the status thresholds that have a normal status value.

**Warning**

Displays the status thresholds that have a warning status value.

**Error**

Displays the status thresholds that have an error status value.

**QUERY status threshold**

Query all status thresholds by issuing the following command:

```
query statusthreshold
```

Threshold Name	Activity Name	Condition Name	Value	Report Status
ACTIVELOGCHECK	ACTIVE LOG UTILIZATION (%)	>	90	ERROR
AVGSTGPLW	AVERAGE STORAGE POOL UTILIZATION (%)	>	85	WARNING
AVGSTGPLE	AVERAGE STORAGE POOL UTILIZATION (%)	>	90	ERROR

## Query status thresholds and display detailed format

Query status thresholds and display the output in detailed format, by issuing the following command:

```
query statusthreshold f=d
```

```
Threshold Name: ACTIVELOGCHECK
Activity Name: ACTIVE LOG UTILIZATION (%)
Condition Name: >
Value: 90
Report Status: ERROR
Server Name: TSMAWP24

Threshold Name: AVGSTGPLW
Activity Name: AVERAGE STORAGE POOL UTILIZATION (%)
Condition Name: >
Value: 85
Report Status: WARNING
Server Name: TSMAWP24

Threshold Name: AVGSTGGLE
Activity Name: AVERAGE STORAGE POOL UTILIZATION (%)
Condition Name: >
Value: 95
Report Status: ERROR
Server Name: TSMAWP24
```

## Related commands

Table 274. Commands related to QUERY STATUSTHRESHOLD

Command	Description
"DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)" on page 326	Defines a status monitoring threshold.
"DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442	Deletes a status monitoring threshold.
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176	Specifies whether to enable client at-risk activity interval evaluation
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)" on page 1180	Specifies the refresh interval for status monitoring.
"SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)" on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
"UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389	Changes the attributes of an existing status monitoring threshold.

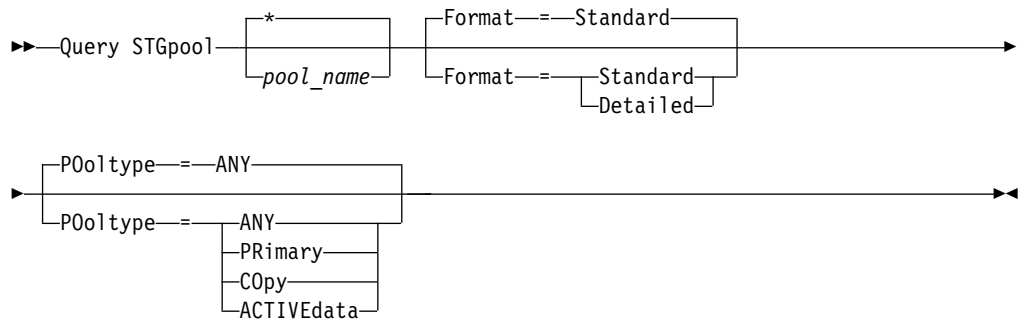
## QUERY STGPOOL (Query storage pools)

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

### 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. Specify one of the following values:

##### Standard

Specifies that partial information is displayed.

##### Detailed

Specifies that complete information is displayed.

#### Pooltype

Specifies the type of storage pool to query. This parameter is optional. The default value is ANY. Specify one of the following values:

##### 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

**Tip:** In the examples of detailed output, some fields are blank because the item does not apply in the specified environment.

Display details for a storage pool that is named DISKPOOL. See “Field descriptions” on page 941 for field descriptions.

```
query stgpool diskpool format=detailed
```

```
Storage Pool Name: DISKPOOL
Storage Pool Type: Primary
Device Class Name: DISK
Storage Type: DEVCLASS
Cloud Type:
Cloud URL:
Cloud Identity:
Cloud Location:
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 Container Reuse: 1 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: 01/03/2014 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?: No
Processes For Identifying Duplicates:
Compressed:
Deduplication Savings:
Compression Savings:
Total Space Saved:
Auto-copy Mode: Client
Contains Data Deduplicated by Client?: No
Maximum Simultaneous Writers:
Protection Storage Pool:
Date of Last Protection:
Deduplicate Requires Backup?:
Encrypted:
Space Utilized(MB):
```

### Example: Display detailed sequential-access disk storage pool information

Display details for a storage pool that is named FILEPOOL. See “Field descriptions” on page 941 for field descriptions.

```
query stgpool filepool format=detailed
```

```

Storage Pool Name: FILEPOOL
Storage Pool Type: Primary
Device Class Name: FILEC
Storage Type: DEVCLASS
Cloud Type:
Cloud URL:
Cloud Identity:
Cloud Location:
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 Container Reuse: 1 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: 01/02/2014 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:
Deduplicate Data?: Yes
Processes For Identifying Duplicates: 1
Compressed:
Deduplication Savings: 65,396 K (49.99%)
Compression Savings:
Total Space Saved: 65,396 K (49.99%)
Auto-copy Mode: Client
Contains Data deduplicated by Client?: Yes
Maximum Simultaneous Writers:
Protection Storage Pool:
Date of Last Protection:
Deduplicate Requires Backup?:
Encrypted:
Space Utilized(MB):

```

### Example: Display detailed sequential storage pool information

Display details for an active-data sequential storage pool that is named FILEPOOL that uses a FILE type device class. See “Field descriptions” on page 941 for field descriptions.

```
query stgpool filepool format=detailed
```

```

Storage Pool Name: FILEPOOL
Storage Pool Type: Active-data
Device Class Name: FILEC
Storage Type: DEVCLASS
Cloud Type:
Cloud URL:
Cloud Identity:
Cloud Location:
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 Container Reuse: 1 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: 01/02/2014 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
Overwrite Data when Deleted:
Deduplicate Data?: Yes
Processes For Identifying Duplicates: 1
Compressed:
Deduplication Savings: 65,396 K (49.99%)
Compression Savings:
Total Space Saved: 65,396 K (49.99%)
Auto-copy Mode:
Contains Data Deduplicated by Client?: No
Maximum Simultaneous Writers:
Protection Storage Pool:
Date of Last Protection:
Deduplicate Requires Backup?:
Encrypted:
Space Utilized(MB):

```

### Example: Display summary information for a specific storage pool

Display information for a storage pool that is named POOL1. See “Field descriptions” on page 941 for field descriptions.

```
query stgpool pool1
```



Storage Pool Name	Device Class Name	Estimated Capacity	Pct Util	Pct Migr	High Mig Pct	Low Mig Pct	Next Storage 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 941 for field descriptions.

```
query stgpool 8mmpool format=detailed
```

```

Storage Pool Name: 8MMPPOOL
Storage Pool Type: Primary
Device Class Name: 8MMTAPE
Storage Type: DEVCLASS
Cloud Type:
Cloud URL:
Cloud Identity:
Cloud Location:
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 Container Reuse: 1 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: 01/08/2014 06:55:45
Storage Pool Data Format: Native
Copy Storage Pool(s): COPYPOOL1
Active Data Pool(s): ACTIVEPOOL1 ACTIVEPOOL2
Continue Copy on Error?: Yes
CRC Data: Yes
Reclamation Type: Threshold
Overwrite Data when Deleted:
Deduplicate Data?: No
Processes For Identifying Duplicates:
Compressed:
Deduplication Savings:
Compression Savings:
Total Space Saved:
Compressed: No
Deduplication Savings:
Compression Savings:
Total Space Saved:
Auto-copy Mode: Client
Contains Data Deduplicated by Client?: No
Maximum Simultaneous Writers:
Protection Storage Pool:
Date of Last Protection:
Deduplicate Requires Backup?:
Encrypted:
Space Utilized(MB):

```

### 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 941 for field descriptions.

```
query stgpool nas2libpool format=detailed
```

```

Storage Pool Name: NAS2
Storage Pool Name: NAS2LIBPOOL
Storage Pool Type: Primary
Device Class Name: NAS2CLASS
Storage Type: DEVCLASS
Cloud Type:
Cloud URL:
Cloud Identity:
Cloud Location:
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 Container Reuse: 1 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: 01/02/2014 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:
Overwrite Data when Deleted:
Deduplicate Data?: No
Processes For Identifying Duplicates:
Compressed:
Deduplication Savings:
Compression Savings:
Total Space Saved:
Auto-copy Mode: Client
Contains Data Deduplicated by Client?: No
Maximum Simultaneous Writers:
Protection Storage Pool:
Date of Last Protection:
Deduplicate Requires Backup?:
Encrypted:
Space Utilized(MB):

```

## 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 that is assigned to the storage pool.

## Estimated Capacity

The estimated capacity of the storage pool in megabytes (M) or gigabytes (G).

For DISK devices, estimated capacity is the capacity of all volumes in the storage pool, including any volumes that are varied offline.

For sequential-access storage pools, estimated capacity is the total estimated space of all the sequential-access volumes in the storage pool, regardless of their access mode. At least one volume must be used in a sequential-access storage pool (either a scratch volume or a private volume) to calculate estimated capacity.

For tape and FILE devices, the estimated capacity for the storage pool includes the following factors:

- The capacity of all the scratch volumes that the storage pool already acquired or can acquire. The number of scratch volumes is defined by the **MAXSCRATCH** parameter on the **DEFINE STGPOOL** or **UPDATE STGPOOL** command.
- The capacity of all the private volumes that are defined to the storage pool when you use the **DEFINE VOLUME** command.
- The estimated capacity of the storage pool cannot be more than the capacity of the total file systems that are used by the device class.

The calculations for estimated capacity depend on the available space of the storage for the device that is assigned to the storage pool. For FILE storage pools, the capacity for the storage pool is reduced if the available storage is less than the total estimated space of all the FILE volumes in the storage pool. The value that is displayed for capacity is reduced by the size of a FILE volume incrementally as the available space continues to decline.

For Centera, value represents the total capacity of the Centera storage device that is being queried.

## Space Trigger Util

Utilization of the storage pool, as calculated by the storage pool space trigger, if any, for this storage pool. You can define space triggers for storage pools that are associated with DISK or FILE device types only.

For sequential access devices, space trigger utilization is expressed as follows as a percentage of the number of used bytes on each sequential access volume relative to the size of the volume and estimated capacity of all existing volumes in the storage pool. It does not include potential scratch volumes. Unlike the calculation for percent utilization, the calculation for space trigger utilization favors creation of new private file volumes by the space trigger over usage of more 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 is on any volumes that are varied offline. The value for space trigger utilization can be higher than the value for percent migration 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 that is allocated while the transaction is in progress. The value for percent migration represents only the space that is 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 must be used for noncached files.

### **Pct Util**

An estimate of the utilization of the storage pool, as a percentage.

For sequential access devices, it is a percentage of the number of active bytes on each sequential access volume and the estimated capacity of all volumes in the storage pool. The percentage includes the number of potential scratch volumes that might be allocated.

For disk devices, it is a percentage of the estimated capacity, including cached data and data that is 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 that is allocated, while the transaction is in progress. The value for **Pct Migr** represents only the space that is 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 must be used for noncached files.

For Centera, this represents an estimate of the utilization of the entire Centera storage device, not the storage pool that is 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. The value includes the capacity of all the scratch volumes that are 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 must 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 that is occupied by client files that might or might 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 that are 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 that is 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 might be stored in the storage pool.

**Access**

The access mode for data in the storage pool. The following access modes are possible:

**Read/Write**

The data can be accessed in read-write mode.

**Read only**

The data can be accessed in read-only mode.

**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 that are 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 File space.

**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 whether reclamation is necessary.

**Offsite Reclamation Limit**

The number of offsite volumes that have space that is 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 that are used in the storage pool.

**Delay Period for Container Reuse (container storage pools only)**

The number of days that must elapse after all files are deleted from a container before the server reuses the container.

**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 is migrated, if migration is in progress. If migration is not in progress, this value indicates the amount of data that was migrated during the last migration. When multiple, parallel migration processes are used for the storage pool, this value indicates the total amount of data that is migrated by all processes.

**Elapsed Migration Time (seconds) (primary storage pools only)**

The amount of time that elapsed since migration began, if migration is active. If migration is not active, this value indicates the amount of time that is 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 is 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 that is used to write data to this storage pool (for example NATIVE, NETAPPDUMP, CELERRADUMP, or NDMPDUMP).

**Copy Storage Pool(s)**

The copy storage pools that are listed have data that is 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 that are listed here have data that is 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 continues to write data to other copy storage pools in the list or ends 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 is deleted from the database.

**Deduplicate Data?**

Whether data in the storage pool is 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.

**Deduplication Savings**

The amount and percentage of data that is saved in the storage pool by using data deduplication.

**Total Space Saved**

The total amount of data that was saved in the storage pool.

**Auto-copy Mode**

Indicates whether data is written simultaneously to copy storage pools or active-data pools during client store sessions, server import processes, server data migration processes, or all three operations. The value CLIENT indicates either client store or server import operations. The value ALL indicates that simultaneous-write operations occur whenever this pool is a target for any of the eligible operations.

If the storage pool is a copy storage pool or an active-data pool or if the simultaneous-write function is disabled, this field is blank.

**Contains Data Deduplicated by Client?**

Indicates whether the storage pool contains data that was deduplicated by clients. Storage pools that contain data that is deduplicated by clients are not accessible for LAN-free data movement by storage agents V6.1 or earlier.



## Related commands

Table 275. Commands related to **QUERY STGPOOL**

Command	Description
	Convert a storage pool to a directory-container storage pool.
<b>COPY ACTIVATEDATA</b>	Copies active backup data.
<b>DEFINE STGPOOL</b>	Defines a storage pool as a named collection of server storage media.
<b>DELETE STGPOOL</b>	Deletes a storage pool from server storage.
	Displays information about storage pool directories.
<b>UPDATE STGPOOL</b>	Changes the attributes of a storage pool.

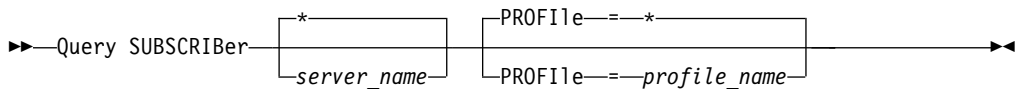
## QUERY SUBSCRIBER (Display subscriber information)

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 a managed server 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.

**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 276. Commands related to QUERY SUBSCRIBER*

<b>Command</b>	<b>Description</b>
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

►► Query SUBSCRIPTION \*  
profile\_name ◀◀

### 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 277. 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.

Table 277. Commands related to **QUERY SUBSCRIPTION** (continued)

<b>Command</b>	<b>Description</b>
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 select statements, SHOW commands, and other Tivoli Storage Manager commands. Output is generated from several Tivoli Storage Manager commands, for example:

- QUERY ASSOCIATION
- QUERY COPYGROUP
- QUERY DATAMOVER
- QUERY DB
- QUERY DBSPACE
- QUERY DEVCLASS
- QUERY DIRSPACE
- QUERY DOMAIN
- QUERY LIBRARY
- QUERY LOG
- QUERY MGMTCLASS
- QUERY OPTION
- QUERY PROCESS
- QUERY REPLRULE
- QUERY SCHEDULE
- QUERY SERVER
- QUERY SESSION
- QUERY STATUS
- QUERY STGPOOL
- QUERY VOLHISTORY
- QUERY VOLUME

### Privilege class

Any administrator can issue this command.

### Syntax

▶—Query SYStem—▶

### Example: View 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 278. 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 DBSPACE	Displays information about the storage space defined for 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 SESSION	Displays information about all active administrator and client sessions with 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 VOLHISTORY	Displays sequential volume history information that has been collected by the server.
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 279. 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)

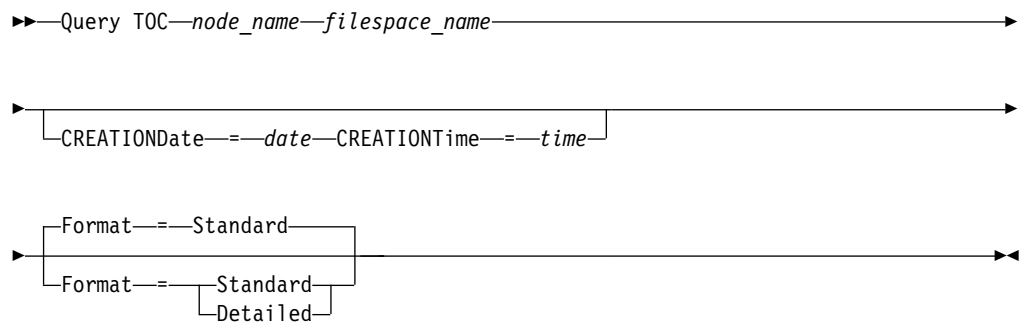
Use this command to display directory and file information contained in the table of contents (TOC) for a specified backup image. This command does not load table of contents information into the IBM Tivoli Storage Manager database. The specified table of contents are read from a storage pool each time 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 is required and output is 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 filesystem /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 280. Commands related to QUERY TOC*

<b>Command</b>	<b>Description</b>
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.

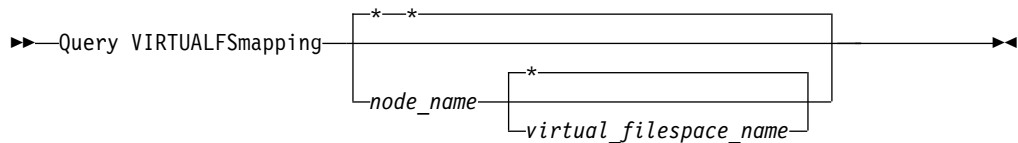
## QUERY VIRTUALFSMAPPING (Query a virtual file space mapping)

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\_file\_space\_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	/nonASCIIDir	/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

The name of the file space that belongs to the node.

File space names can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

**Path** Specifies the path to the client node.

**Hexadecimal Path**

Indicates whether the path is hexadecimal.

**Related commands**

*Table 281. 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.

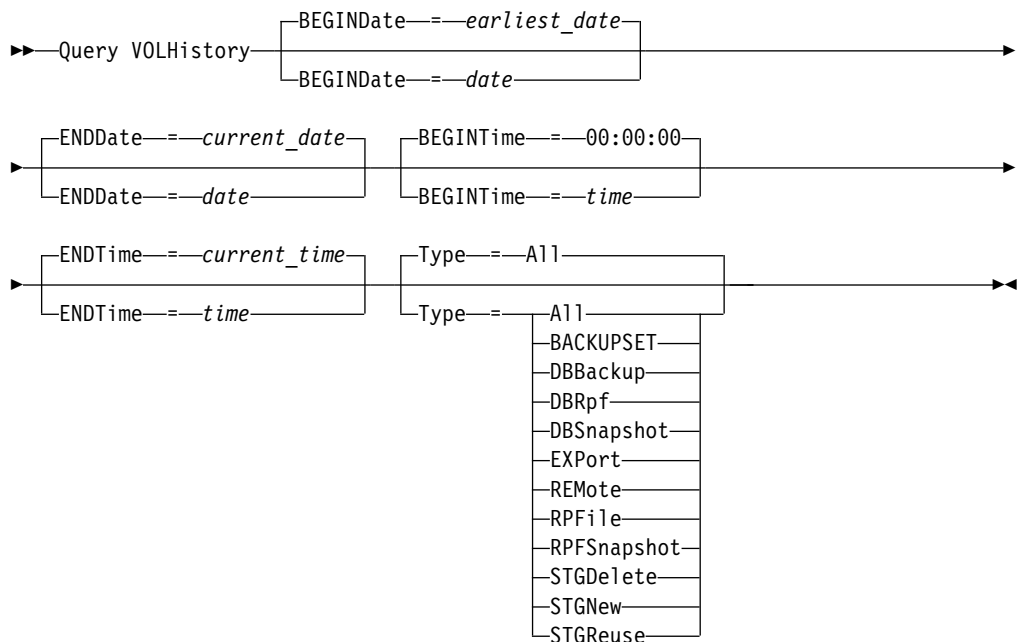
Use the VOLUMEHISTORY server option to specify one or more volume history files. After the server is restarted, Tivoli Storage Manager updates volume information in both the database and the files.

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

Value	Description	Example
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified	TODAY-7 or -7.  To display information beginning with records created a week ago, specify BEGINDATE=TODAY-7 or BEGINDATE=-7.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### 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- <i>days</i> or - <i>days</i>	The current date minus days specified. The maximum number of days is 9999.	TODAY-1 or -1.  To display records created up to yesterday, specify ENDDATE=TODAY-1 or ENDDATE=-1.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

#### 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
<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> or + <i>HH:MM</i>	The current time plus hours and minutes on the specified begin date	NOW+03:00 or +03:00. 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- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes on the specified begin date	NOW-03:30 or -03:30. 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 + <i>HH:MM</i>	The current time plus hours and minutes on the specified end date	NOW+03:00 or +03:00. If you issue this command at 9:00 with ENDTIME=NOW+03:00 or ENDTIME=+03:00, Tivoli Storage Manager displays records with a time of 12:00 or later on the end date.
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes on the specified end date	NOW-03:30 or -03:30 If you issue this command at 9:00 with ENDTIME=NOW-3:30 or ENDTIME=-3:30, Tivoli Storage Manager displays records with a time 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 file objects of a recovery plan that are saved on a target server and that were created assuming database full and incremental backups. The parameter displays only records about recovery plan files that are saved on another Tivoli Storage Manager server by using the server-to-server virtual volume function for Tivoli Storage Manager.

**RPFSnapshot**

Specifies to display only records that contain information about file objects of a recovery plan that are saved on a target server and that were created assuming database snapshot backups. **RPFSnapshot** only displays records about recovery plan files that are saved on another Tivoli Storage Manager server by using the server-to-server virtual volume function for Tivoli Storage Manager.

**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 for a storage pool volume**

Display volume history information for a storage pool volume stored in the Tivoli Storage Manager database. See "Field descriptions" on page 965 for field descriptions. Issue the command:

```
query volhistory type=stgnew
```

```

Date/Time: 02/25/2011 18:28:06
Volume Type: STGNEW
Backup Series:
Backup Operation:
Volume Seq:
Device Class: FILE
Volume Name: /adsmfct/server/prv011
Volume Location:
Command:
Database Backup ID High:
Database Backup ID LOW:
Database Backup Home Position:
Database Backup HLA:
Database Backup LLA:
Database Backup Total Data Bytes (MB):
Database Backup total Log Bytes (MB):
Database Backup Block Num High:
Database Backup Block Num Low:
Database Backup Stream Id:
Database Backup Volume Sequence for Stream:

```

**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.

### Example: Display volume history information for a database backup volume

Display volume history information for a database backup volume stored in the Tivoli Storage Manager database. See “Field descriptions” on page 965 for field descriptions. Issue the command:

```
query volhistory type=dbb
```

```

Date/Time: 02/25/2011 18:28:06
Volume Type: BACKUPFULL
Backup Series: 176
Backup Operation: 0
Volume Seq: 0
Device Class: FILE
Volume Name: /adsmfct/server/prv011
Volume Location:
Command:
Database Backup ID High: 0
Database Backup ID LOW: 0
Database Backup Home Position: 0
Database Backup HLA:
Database Backup LLA:
Database Backup Total Data Bytes (MB): 0
Database Backup total Log Bytes (MB): 0
Database Backup Block Num High: 0
Database Backup Block Num Low: 0
Database Backup Stream Id: 1
Database Backup Volume Sequence for Stream: 10,001

```

**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.

#### **RPFSnapshot**

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 RPFSNAPSHOT volume type: the identifier of the backup series that is associated with the RPFSNAPSHOT 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.

**Note:** The following fields are not used by Tivoli Storage Manager servers that are V6.3 or later. However, the fields are displayed for compatibility with earlier releases.

- Database Backup ID High
- Database Backup ID Low
- Database Backup Home Position
- Database Backup HLA
- Database Backup LLA
- Database Backup Total Data Bytes (MB)
- Database Backup Total Log Bytes (MB)
- Database Backup Block Num High
- Database Backup Block Num Low

## Related commands

*Table 282. 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.
QUERY BACKUPSET	Displays backup sets.
UPDATE VOLHISTORY	Adds or changes location information for a volume in the volume history file.

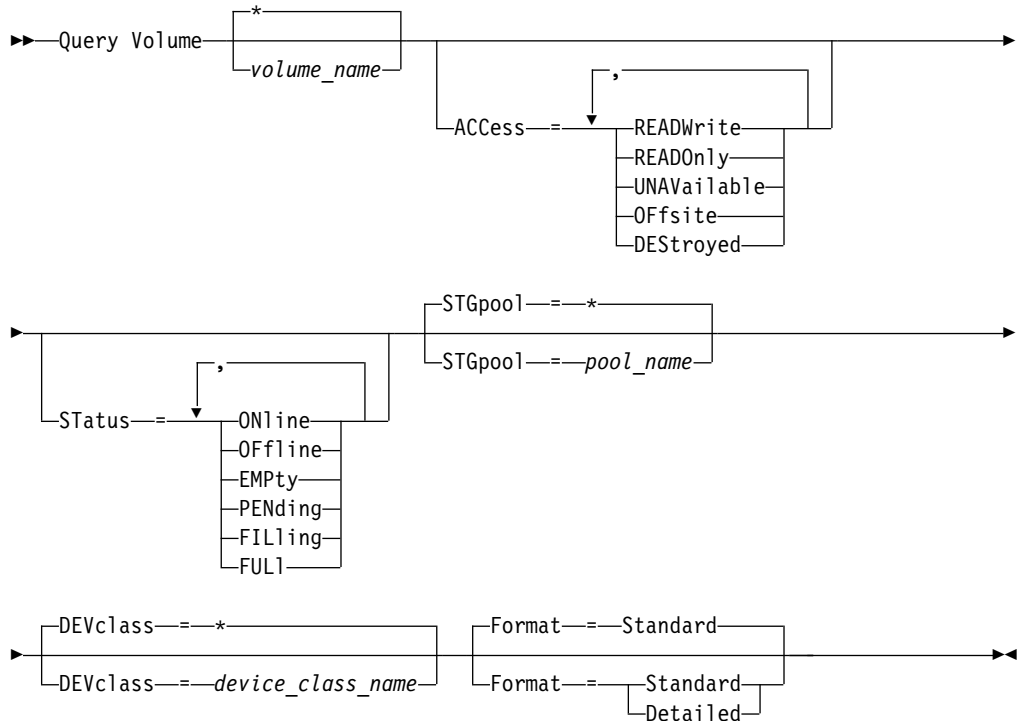
## QUERY VOLUME (Query storage pool volumes)

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 read only files that are stored on the volumes.

#### **UNAVailable**

Display volumes with an access mode of UNAVAILABLE. Client nodes and server processes cannot access files that are 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 are 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 volumes might be sequential-access volumes from which all files were deleted, but for which the time specified by the **REUSEDELAY** parameter on the **DEFINE STGPOOL** command has not elapsed. These volumes might 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 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 972 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 972 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:
Logical Block Protected:
Drive Encryption Key Manager:
```

## Example: Display detailed information about a storage pool volume 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 972 for field descriptions.

```
query volume devclass=fileclass format=detailed
```



```

        Volume Name: /WORM_FILESYS/0000000E.BFS
        Storage Pool Name: FILEPOOL
        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
        Logical Block Protected:
        Drive Encryption Key Manager:

```

### Example: Display detailed information about a specific storage pool volume

Display details about a storage pool volume that is named 000642. The volume is in a storage pool that is associated with a 3592 device class. See “Field descriptions” on page 972 for field descriptions.

```
query volume 000642 format=detailed
```

```

        Volume Name: 000642
        Storage Pool Name: 3592POOL
        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
        Logical Block Protected: Yes
        Drive Encryption Key Manager: Tivoli Storage Manager

```

## Field descriptions

### Volume Name

The name of the storage pool volume.

### Storage Pool Name

The storage pool to which the volume is defined.

### Device Class Name

The device class that is 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, which is 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 store only 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 that is occupied by both files and aggregates, including empty space within aggregates.

For DISK volumes, the utilization also includes space that is 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 whether reclamation is necessary. Reclaimable space includes empty space within aggregates.

When determining which volumes in a storage pool to reclaim, the server first determines the reclamation threshold. The reclamation threshold is indicated by the value of the **THRESHOLD** parameter on the **RECLAIM STGPOOL** command or, if that value was not specified, the value of the **RECLAIM** parameter in a storage pool definition. The server then examines the percentage of reclaimable space for each volume in the storage pool. If the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool, the volume is a candidate for reclamation.

For example, suppose that storage pool FILEPOOL has a reclamation threshold of 70 percent. This value indicates that the server can reclaim any volume in the storage pool that has a percentage of reclaimable space that is greater than 70 percent. The storage pool has three volumes:

- FILEVOL1 with 65 percent reclaimable space
- FILEVOL2 with 80 percent reclaimable space
- FILEVOL3 with 95 percent reclaimable space

When reclamation begins, the server compares the percent of reclaimable space for each volume with the reclamation threshold of 70 percent. In this example, FILEVOL2 and FILEVOL3 are candidates for reclamation because their percentages of reclaimable space are greater than 70.

For volumes that belong to a SnapLock storage pool, the value is displayed but is not used.

**Scratch Volume? (sequential access volumes only)**

Whether this volume is 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 Tivoli Storage Manager.

**Number of Times Mounted**

The number of times that the server opened the volume for use. The number of times that the server opened the volume is not always the same as the number of times that the volume was 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 was 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 was changed to pending.

**Number of Write Errors**

The number of writing errors that occurred on the volume.

**Number of Read Errors**

The number of reading errors that 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 was defined and used (at least once) by the Tivoli Storage Manager 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 server begins 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 are 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 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 that is associated with a device type of 3592, LTO, or ECARTRIDGE.

#### **Logical Block Protected**

Specifies whether logical block protection is enabled for the volume. You can use logical block protection only with the following types of drives and media:

- IBM LTO5 and later
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later
- Oracle StorageTek T10000C and T10000D drives

### **Related commands**

*Table 283. Commands related to QUERY VOLUME*

<b>Command</b>	<b>Description</b>
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 does not move inactive versions of backup data from volumes in active-data pools.

This command cannot be used for the following types of storage pools:

- Storage pools with one of the following data formats:
  - NETAPPDUMP
  - CELERRADUMP
  - NDMPDUMP
- Storage pools that use a CENTERA device class.
- Storage pools that use a Write Once Read Many (WORM) device class. Reclamation is not necessary because WORM volumes are not reusable, but you can run reclamation to consolidate data onto fewer volumes.

Use this command only if you are not going to use automatic reclamation for the storage pool. This command accepts the values of the **RECLAIMPROCESS** and **RECLAIMSTGPOOL** attributes of the storage pool definition. This command also accepts the values of the **OFFSITERECLAIMLIMIT** and **RECLAIM** parameters of the storage pool definition, if not overridden by the **OFFSITERECLAIMLIMIT** and **THRESHOLD** command parameters.

### Tips:

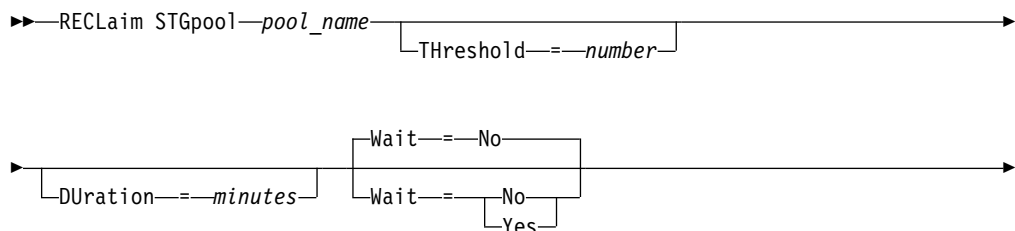
- When you issue this command, duplicate data in a primary storage pool, copy storage pool, or active-data pool that is set up for data deduplication is removed.
- When you use this command to restore deduplicated objects to the same storage pool, any duplicate data blocks are replaced with references to deduplicated extents.

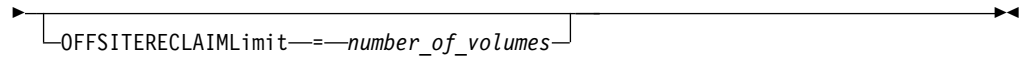
For storage pools defined with **RECLAMATIONTYPE=SNAPLOCK**, this command also deletes empty WORM FILE volumes that exceeded their reclaim period.

### Privilege class

To issue this command, you must have system privilege, unrestricted storage privilege, or restricted storage privilege for the storage pool that is being reclaimed and the reclaim storage pool, if applicable.

### Syntax





## Parameters

### *pool\_name* (Required)

Specifies the storage pool in which volumes are to be reclaimed.

### **D**uration

Specifies the maximum number of minutes that the reclamation runs before it is automatically canceled. You can specify a number 1 - 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 that is being reclaimed. The server also checks the reclamation process 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 the duration period 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 might cause the reclamation to end before any volumes are reclaimed. In most situations when you initiate reclamation for a copy storage pool with offsite volumes, consider limiting the number of offsite volumes to be reclaimed rather than limiting the duration. For details, see the **OFFSITERECLAIMLIMIT** parameter.

### **T**hreshold

Specifies the percentage of reclaimable space on a volume that makes it eligible for reclamation. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the server database. Reclaimable space also includes unused space.

You can specify a number 1 - 99. This parameter is optional. If not specified, the **RECLAIM** attribute of the storage pool definition is used.

To determine the percentage of reclaimable space for a volume, issue the **QUERY VOLUME** command and specify **FORMAT=DETAILED**. The value in the field Pct. Reclaimable Space is the percentage of reclaimable space for the volume.

Specify a value of 50 percent or greater for this parameter so that files stored on two volumes can be combined into a single target volume.

### **OFFSITERECLAIMLimit**

Specifies the maximum number of offsite storage pool volumes that the server tries to reclaim. This parameter is valid only for copy storage pools. You can specify a number 0 - 99999. This parameter is optional. If not specified, the **OFFSITERECLAIMLIMIT** attribute of the storage pool definition is used.

**Wait**

Specifies whether to wait for the server to complete processing this command in the foreground. This parameter is optional. The default is NO. You can specify one of the following values:

**No** Specifies that the server processes this command in the background.

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

If you cancel this process, some files might already be moved to new volumes before the cancellation.

**Yes**

Specifies that the server processes this command in the foreground. The operation must complete before you can continue with other tasks. Output messages are displayed 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.

**Restriction:** 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 reclamation ends as soon as possible after 60 minutes.

```
reclaim stgpool tapepool duration=60
```

**Related commands**

*Table 284. Commands related to RECLAIM STGPOOL*

Command	Description
CANCEL PROCESS	Cancels a background server process.
MIGRATE STGPOOL	Migrates files from a primary storage pool to the next storage pool in the hierarchy.
MOVE DRMEDIA	Moves DRM media onsite and offsite.
QUERY DRMEDIA	Displays information about disaster recovery volumes.
QUERY PROCESS	Displays information about background processes.
QUERY STGPOOL	Displays information about storage pools.



---

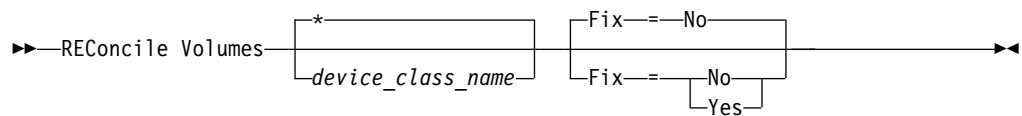
## RECONCILE VOLUMES (Reconcile differences in the virtual volume definitions)

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.

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 <b>Storage pool volumes:</b> Marked as unavailable
		Files exist but marked for deletion	Report error <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 <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 285. 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.
UPDATE SERVER	Updates information about a server.



---

## REGISTER commands

Use the **REGISTER** commands to define or add objects to Tivoli Storage Manager.

- “REGISTER ADMIN (Register an administrator ID)” on page 983
- “REGISTER LICENSE (Register a new license)” on page 987
- “REGISTER NODE (Register a node)” on page 989

## REGISTER ADMIN (Register an administrator ID)

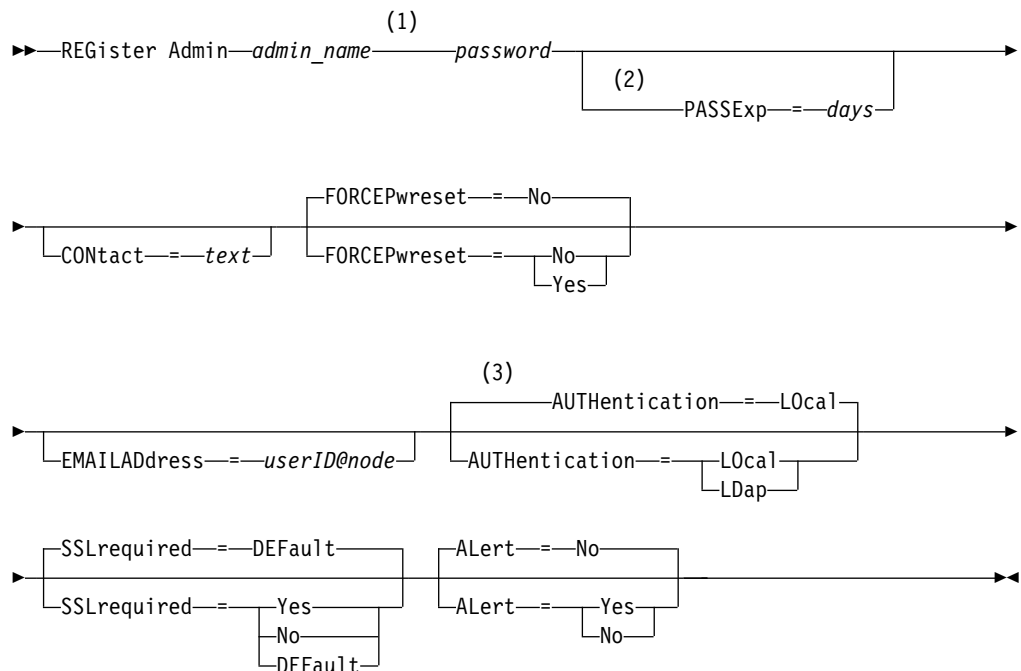
Use this command to add an administrator ID to the server. After registration, the administrator ID can issue a limited set of commands, including all query commands. To provide additional privileges, use the **GRANT AUTHORITY** command.

### Privilege class

To issue this command, you must have system privilege.

**Remember:** When you register an administrator with the same name as an existing node, be aware of the administrator authentication method and the **SSLREQUIRED** setting. Any node that has the same name as the administrator that is being registered inherits those settings.

### Syntax



### Notes:

- 1 A password is not required if you register an administrator and select **AUTHENTICATION=LDAP**. At logon, you are prompted for a password.
- 2 The **PASSEXP** command does not apply to administrators who authenticate to an LDAP directory server.
- 3 The default value can change if you issued the **SET DEFAULTAUTHENTICATION** command and specified LDAP.

### Parameters

#### *admin\_name* (Required)

Specifies the name of the administrator to be registered. The maximum length of the name is 64 characters.

You cannot specify an administrator name of NONE.

*password* **(Required)**

Specifies the password of the administrator to be registered. The maximum length of the password is 64 characters. The password is not case-sensitive if authentication is with the Tivoli Storage Manager server. Passwords that authenticate with an LDAP directory server are case-sensitive. See “Naming Tivoli Storage Manager objects” on page 13 for the characters that are available for specifying a password.

**Tip:** A password is not required if you register an administrator and select AUTHENTICATION=LDAP. At logon, you are prompted for 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 password is set with the global expiration period of 90 days. This parameter does not affect passwords that authenticate with an LDAP directory server.

**CONTACT**

Specifies information identifying the administrator being registered. This parameter is optional. The maximum length of this string is 255 characters. The contact information must be enclosed in quotation marks if it contains any blanks.

**FORCEPwreset**

Specifies whether the administrator is required to change or reset the password. This parameter is optional. The default value is NO. 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.

**Yes**

Specifies that the administrator's password expires at the next sign-on. The client or administrator must change or reset the password then. If a password is not specified, you receive an error message.

**EMAILAddress**

Specifies the email address for this administrator.

**AUTHentication**

This parameter specifies the authentication method for the administrator user ID. Specify one of the following values: LDAP or LOCAL. The parameter is optional and defaults to LOCAL. The default can change to LDAP if you use the **SET DEFAULTAUTHENTICATION** command and specify LDAP.

**Local**

Specifies that the local Tivoli Storage Manager server database is used.

**LDap**

Specifies that the administrator user ID authenticates passwords with an LDAP directory server. Passwords that authenticate with an LDAP directory server are case-sensitive.

**Tip:** A password is not required if you register an administrator and select AUTHENTICATION=LDAP. At logon, you are prompted for a password.

**SSLrequired**

Specifies whether the administrator user ID must use Secure Sockets Layer (SSL) to communicate between the Tivoli Storage Manager server and the

backup-archive client. When you authenticate passwords with an LDAP directory server, you must protect the sessions by using SSL or another network security method.

**Yes**

Specifies that SSL is required.

**No** Specifies that SSL is not required.

**DEFAuLt**

Specifies that SSL is required for an administrator if the password that is associated with its user ID authenticates with an LDAP directory server. SSL is not required for an administrator ID that authenticates its password with the Tivoli Storage Manager server (LOCAL).

**ALert**

Specifies whether alerts are sent to an administrators email address.

**Yes**

Specifies that alerts are sent to the specified administrators email address.

**No** Specifies that alerts are not sent to the specified administrators email address. This is the default value.

**Tip:** Alert monitoring must be enabled, and email settings must be correctly defined to successfully receive alerts by email. To view the current settings, issue the **QUERY MONITORSETTINGS** command.

**Example: Register an administrator ID**

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'
```

**Example: Register an administrator ID and set the authentication method**

Define an administrator ID for Harry, use the password Pa##\$twO. Using the ID and password, Harry can authenticate to the LDAP directory server. Issue the command:

```
register admin Harry Pa##$tw0 authentication=ldap
```

If the password specified does not adhere to LDAP password standards, the administrator is registered, and at log-in a new password must be entered.

**Related commands**

*Table 286. Commands related to REGISTER ADMIN*

Command	Description
GRANT AUTHORITY	Assigns privilege classes to an administrator.
LOCK ADMIN	Prevents an administrator from accessing Tivoli Storage Manager.
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.

Table 286. Commands related to REGISTER ADMIN (continued)

Command	Description
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
REGISTER NODE	Defines a client node to the server and sets options for that user.
REMOVE ADMIN	Removes an administrator from the list of registered administrators.
RENAME ADMIN	Changes a Tivoli Storage Manager administrator's name.
SET DEFAULTAUTHENTICATION	Specifies the default password authentication method for any REGISTER NODE or REGISTER ADMIN commands.
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 Tivoli Storage Manager.
UPDATE ADMIN	Changes the password or contact information associated with any administrator.
UPDATE NODE	Changes the attributes that are 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 in the server instance directory of your installation, and reregister any previously registered licenses.
- You cannot register licenses for Tivoli Storage Manager for Mail, Tivoli Storage Manager for Databases, Tivoli Storage Manager for ERP, and Tivoli Storage Manager for Space Management.

To generate a report that can help you understand the license requirements for your system, run the **QUERY PVUESTIMATE** command. The report contains estimates of the number of client devices and PVU totals for server devices. The estimates are not legally binding.

### 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. The following values can be used:

#### **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).

**\*.lic**

To license all IBM Tivoli Storage Manager licenses for server components.

**Example: Register a license**

Register the base Tivoli Storage Manager license.

```
register license file=tsmbasic.lic
```

**Related commands**

*Table 287. Commands related to REGISTER LICENSE*

<b>Command</b>	<b>Description</b>
AUDIT LICENSES	Verifies compliance with defined licenses.
QUERY LICENSE	Displays information about licenses and audits.
QUERY PVUESTIMATE	Displays processor value unit estimates.
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 exists with the same name as the node that is 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.

If you do not specify an administrator name or select **USERID=NONE**, the administrator is given the same name as the node. To keep the same-named node and administrator synchronized, the authentication method and the **SSLREQUIRED** setting for the administrator are updated to match the node. Passwords that are shared between same-named nodes and administrators are kept synchronized during an authentication change.

If you register a node that has the same name as an administrator, the administrator authentication method and **SSLREQUIRED** setting change to match the method and authentication method of the node.

**Important:** When you set `sslrequired=serveronly` in a **REGISTER NODE** command, the admin **SSLREQUIRED** setting reverts to YES. To use a non-SSL session with a storage agent, rename the admin with the identical name by issuing the **RENAME ADMIN** command.

When you register or update a node, you can specify whether damaged files on the node can be recovered from a replication server. Files can be recovered only if all the following conditions are met:

- Tivoli Storage Manager, Version 7.1.1 or later, is installed on the source and target replication servers.
- The **REPLRECOVERDAMAGED** system parameter is set to ON. The system parameter can be set by using the **SET REPLRECOVERDAMAGED** command.
- The source server includes at least one file that is marked as damaged in the node that is being replicated.
- The node data was replicated before the damage occurred.

The following table describes how parameter settings affect the recovery of damaged, replicated files.

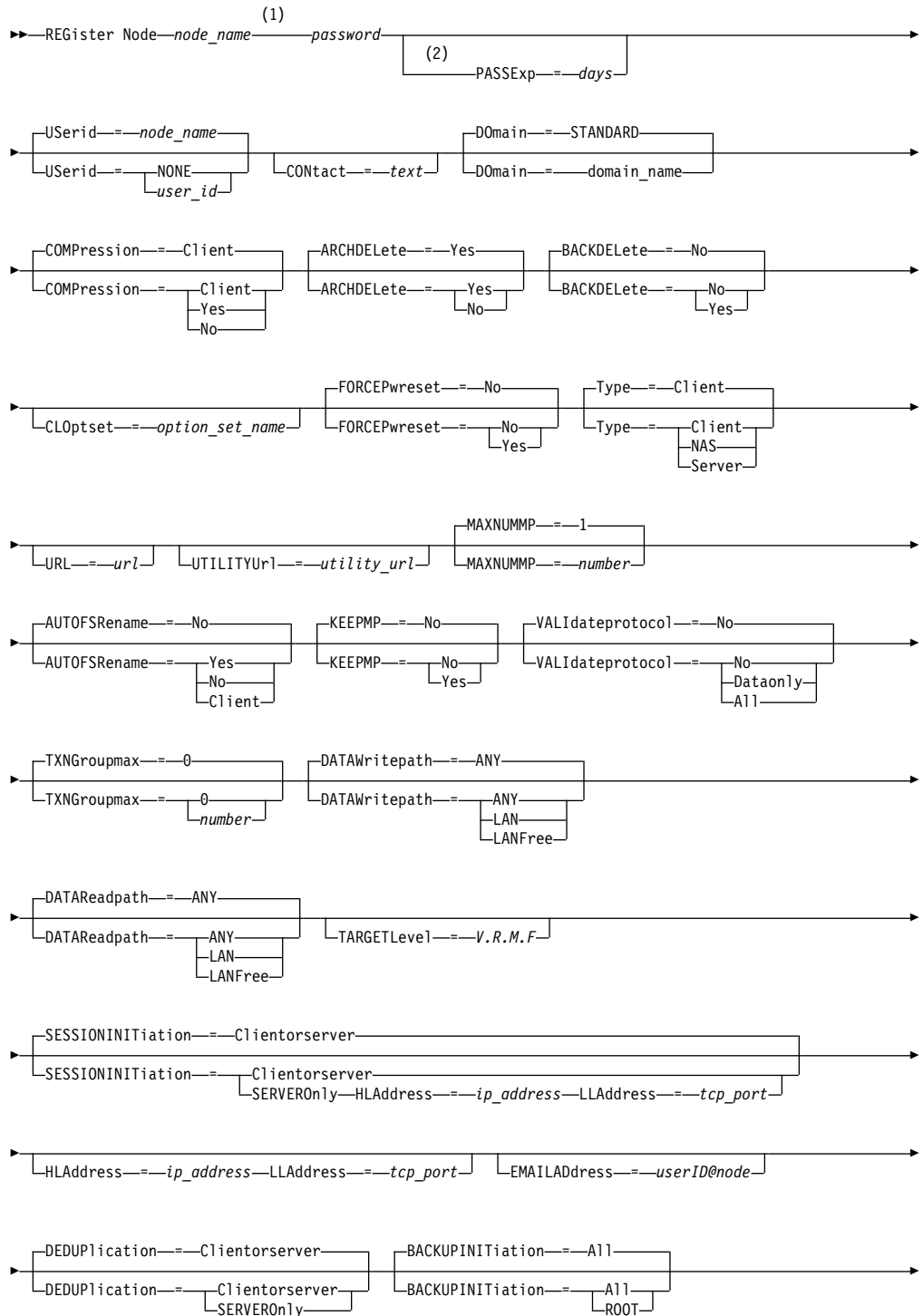
Table 288. Settings that affect the recovery of damaged files.

Setting for the <b>REPLRECOVERDAMAGED</b> system parameter	Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REPLICATE NODE</b> command	Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REGISTER NODE</b> and <b>UPDATE NODE</b> commands	Result
OFF	YES, NO, or not specified	YES or NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.
OFF	ONLY	YES or NO	An error message is displayed because files cannot be recovered when the <b>REPLRECOVERDAMAGED</b> system parameter is set to OFF.
ON	YES	YES or NO	During node replication, standard replication occurs and damaged files are recovered from the target replication server.
ON	NO	YES or NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.
ON	ONLY	YES or NO	Damaged files are recovered from the target replication server, but standard node replication does not occur.
ON	Not specified	YES	During node replication, standard replication occurs and damaged files are recovered from the target replication server.
ON	Not specified	NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.

## Privilege class

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

## Syntax





### Notes:

- 1 A password is not required if you register an administrator and select AUTHENTICATION=LDAP. At logon, you are prompted for a password.
- 2 The **PASSEX** command does not apply to administrators who authenticate with a Lightweight Directory Access Protocol (LDAP) directory server.
- 3 You can specify the **BKREPLRULEDEFAULT**, **ARREPLRULEDEFAULT**, or **SPREPLRULEDEFAULT** parameter only if you specify the **REPLSTATE** parameter.
- 4 The default value can change if you issued the **SET DEFAULTAUTHENTICATION** command and specified LDAP.

### 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 password, which has a maximum length of 64 characters.

A password that authenticates with the Tivoli Storage Manager server is considered "LOCAL" and is not case-sensitive. A password that authenticates

with an LDAP server is considered "LDAP" and is case-sensitive. See "Naming Tivoli Storage Manager objects" on page 13 for a table that lists the characters available for specifying a password.

**Tip:** A password is not required if you register a node and select AUTHENTICATION=LDAP. At logon, you are prompted for a password.

#### **PASSExp**

Specifies the number of days the password remains valid. You can set the password expiration period 0 - 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 common-password expiration period is used. The common password expiration period is 90 days unless changed by issuing the **SET PASSEXP** command.

You can change the password expiration period by using the **UPDATE NODE** or **SET PASSEXP** commands. You can issue the **SET PASSEXP** command to set a common expiration period for all administrators and client nodes. You can also use the command 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 by using the **SET PASSEXP** command.

You can use the **RESET PASSEXP** command to reset the password expiration period to the common expiration period. The **PASSEXP** command does not apply to nodes that authenticate with an LDAP server.

#### **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 by using the node name 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. You can specify one of the following values:

##### **NONE**

Specifies that an administrative user ID is not automatically defined.

##### *user\_id*

Specifies a node ID different from the node that is 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 that identifies 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 that is specified in the archive copy group of the default management class of that domain.

**COMPression**

Specifies whether the client node compresses its files before it sends these files to the server for backup and archive. The parameter is optional. The default value is CLIENT.

**Note:** This parameter does not apply to nodes with a type of NAS or SERVER. You can specify one of the following values:

**Client**

Specifies that the client determines whether to compress files.

**Yes**

Specifies that the client node compresses its files before it sends these files to the server for backup and archive.

**No** Specifies that the client node does not compress its files before it sends these files 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. You can specify one of the following values:

**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. The parameter is optional. The default value is NO. You can specify one of the following values:

**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. You can specify one of the following values:

**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 the client is logging on to the server.

**Yes**

Specifies that the client node password expires at the next logon. The client must change or reset the password then. If a password is not specified, you receive an error message.



**Type**

Specifies the type of node that is being registered. The parameter is optional. The default value is CLIENT. You can specify one of the following values:

**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 is protected by 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 that is being registered on the target server.

**URL**

Specifies the URL of the Tivoli Storage Manager web client that is configured on the client system. You can use the URL in a web browser and in the Operations Center to remotely manage the client node.

This parameter is optional. The URL must include the DNS name or IP address of the client system, and the port number that is defined on the client system for the Tivoli Storage Manager web client. For example, `http://client.mycorp.com:1581`

**UTILITYURL**

Specifies the address of the Tivoli Storage Manager client management services that are configured on the client system. This URL is used by the Operations Center to access client log files so that you can remotely diagnose client issues from the Operations Center.

This parameter is optional. You can specify a URL of up to 200 characters in length. The URL must start with `https`. It includes the DNS name or IP address of the client system, and the port number that is defined on the client system for the Tivoli Storage Manager client management services. For example, `https://client.mycorp.com:9028`

If you omit the port number, the Operations Center uses the port number 9028, which is the default port number when you install the client management services on the client system.

**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 might prevent the data store operations from being able to acquire mount points.

For volumes in a storage pool that is associated with the FILE or CENTERA device type, the server can have multiple sessions to read and one process to write to the same volume concurrently. To increase concurrency and provide efficient access for nodes with data in FILE or CENTERA storage pools, increase the value of the **MAXNUMMP** parameter.

For nodes that store data into primary storage pools with the simultaneous-write function that is enabled, you must adjust the value of the **MAXNUMMP** parameter to specify the correct number of mount points for each client session. A client session requires one mount point for the primary storage pool and one mount point for each copy storage pool and each active-data pool.

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 one. Otherwise, you receive an error.

A storage agent independently tracks the number of points that are used during a client session. If a node has a storage agent that is installed, it might exceed the **MAXNUMMP** value. The **MAXNUMMP** value might also be exceeded under conditions where the node does not have to wait for a mount point.

**Note:** The server might preempt a client operation for a higher priority operation and the client might lose a mount point if no other mount points are available.

#### **KEEPMP**

Specifies whether the client node keeps the mount point for the entire session. The parameter is optional. The default value is NO. You can specify one of the following values:

##### **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 the data is stored to a sequential access storage pool, any mount points held by the session will not be released.

##### **No**

Specifies that the client node releases the mount point during the session. If policy definitions cause data to be stored to a disk storage pool after the data is stored 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 whether file spaces are renamed by the client, if needed. The parameter is optional. 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 by 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 by using the UTF-8 code page. UTF-8 is a byte-oriented encoding form that is specified by the Unicode Standard.

You can specify one of the following values:

**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).

**VALIDateprotocol**

Specify whether Tivoli Storage Manager completes a cyclic redundancy check (CRC) to validate the data that is sent between the client and server. The parameter is optional. The default is NO. You can specify one of the following values:

**No** Specifies that data validation is not completed on any data that is sent between the client and server.

**Dataonly**

Specifies that data validation is completed only on file data that is sent between the client and server. This does not include the file metadata. This mode impacts performance because extra effort is required to calculate and compare CRC values between the client and the server.

**All**

Specifies that data validation is completed 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 extra effort 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. The parameter is optional. Client performance might be improved by using a larger value for this option.

The default value is 0. Specifying 0 indicates that the node uses 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 increases the recovery log usage. Higher recovery log usage might increase the risk of running out of log space. Evaluate the performance of each node before you change the parameter.

#### **DATAWritepath**

Specifies the transfer path that is 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 fails.

You can specify one of the following values:

##### **ANY**

Specifies that data is sent to the server, storage agent, or both, by any available path. A LAN-free path is used if one is available. If a LAN-free path is unavailable, the data is moved by using the LAN.

##### **LAN**

Specifies that data is sent by using the LAN.

##### **LANFree**

Specifies that data is sent by using a LAN-free path.

#### **DATAReadpath**

Specifies the transfer path that is 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 fails. The value for the transfer path also applies to failover connections. If the value is set to LANFree, failover cannot occur for the node on the secondary server.

You can specify one of the following values:

##### **ANY**

Specifies that the server, storage agent, or both use any available path to read data. A LAN-free path is used if one is available. If a LAN-free path is unavailable, the data is read by using the LAN.

##### **LAN**

Specifies that data is read by using the LAN.

##### **LANFree**

Specifies that data is read by using a LAN-free path.

#### **TARGETLevel**

Specifies the client deployment package that is targeted for this node. You can substitute an applicable release package for Version.Release.Modification.Fix (V.R.M.F) Level. For example: TARGETLevel=6.2.0.0.

You must specify each segment with a number that is applicable to a deployment package. You cannot use an asterisk in any field as a substitution for a valid number. The parameter is optional.

**Restriction:** The **TARGETLEVEL** parameter does not apply to nodes with a type of NAS or SERVER.

### **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 might initiate sessions with the server by communicating on the TCP/IP port that is defined with the server option TCPPOINT. Server-prompted scheduling might also be used to prompt the client to connect to the server.

#### **SERVEROnly**

Specifies that the server does not accept client requests for sessions. All sessions must be initiated by server-prompted scheduling on the port that is 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 parameter must be used when SESSIONINITIATION is set to SERVERONLY, regardless of any addresses that are previously used by the client to contact the server.

The address can be specified either in numeric or host name format. If a numeric address is used, it is 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 are verified with a domain name server. Verified names are 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 parameter must be used when SESSIONINITIATION is set to SERVERONLY, regardless of any addresses that are previously used by the client to contact the server.

The value for this parameter must match the value of client option TCPCLIENTPORT. The default value is 1501.

#### **EMAILAddress**

This parameter is used for more contact information. The parameter is optional. The information that is specified by this parameter is not acted upon by Tivoli Storage Manager.

#### **DEDUPLICATION**

Specifies where data deduplication can occur for this node. The parameter is optional. You can specify one of the following values:

#### **Clientorserver**

Specifies that data stored by this node can be deduplicated on either the client or the server. This value is the default. For data deduplication to take

place on the client, you must also specify a value of YES for the DEDUPLICATION client option. You can specify this option in the client option file or in the client option set on the Tivoli Storage Manager server.

**SERVEROnly**

Specifies that data stored by this node can be deduplicated on the server only.

**BACKUPINITiation**

Specifies whether the non-root user ID on the client node can back up files to the server. The parameter is optional. The default value is ALL, indicating that non-root user IDs can back up data to the server. You can select one of the following values:

**All**

Specifies that non-root user IDs can back up files to the server. ALL is the default if BACKUPINITIATION is not specified.

**ROOT**

Specifies that the root user ID can back up files to the server. If you are using the V6.4 or later backup-archive client, authorized users have the same privileges as the root user ID.

**Restriction:** The attribute is ignored by the server if the backup-archive client connects from an operating system other than AIX, HP-UX, Linux, Solaris, or Mac OS.

**Remember:** The application programming interface (API) is affected by the **BACKUPINITIATION** parameter on the server. By default, all API users are allowed to back up data. Setting the parameter to ROOT on an API node is not recommended.

**REPLState**

Specifies whether data that belongs to the client node is ready to be replicated. This parameter is optional. Specify this parameter only if you are issuing the **REGISTER NODE** command on a server that is configured to replicate data to a target replication server. If you register a client node on a source replication server and set up replication for the node, do not register the node on the target replication server. The client node is created automatically on the target server the first time that replication occurs.

You can select one of the following values:

**ENabled**

Specifies that the client node is configured for replication and is ready to replicate. When you specify this parameter, the replication mode in the client node definition on the source replication server is automatically set to SEND. This setting indicates that data that belongs to the client node is sent to a target server during replication.

When replication first occurs for the client node, the replication state of the node on the target replication server is automatically set to ENABLED. The replication mode on the target replication server is set to RECEIVE. This setting indicates that data that belongs to the client node is received from a source replication server. To determine the replication state and mode, issue the **QUERY NODE** command on a source or a target replication server.

**DISabled**

Specifies that the node is configured for replication but that replication does not occur until you enable it.

**BKREPLRuledefault, ARREPLRuledefault, and SPREPLRuledefault**

Specifies the replication rule that applies to a data type if the file space rules for the data type are set to DEFAULT.

**Restriction:** You can specify the **BKREPLRULEDEFAULT**, **ARREPLRULEDEFAULT**, or **SPREPLRULEDEFAULT** parameter only if you specify the **REPLSTATE** parameter.

**BKREPLRuledefault**

Specifies the replication rule for backup data.

**ARREPLRuledefault**

Specifies the replication rule for archive data.

**SPREPLRuledefault**

Specifies the replication rule for space-managed data.

If the file space rules for the data type are set to DEFAULT and you do not specify a rule for the **BKREPLRULEDEFAULT**, **ARREPLRULEDEFAULT**, or **SPREPLRULEDEFAULT** parameter, data is replicated according to the server rule for the data type.

You can specify normal-priority replication or high-priority replication rules. In a replication process that includes both normal and high-priority data, high-priority data is replicated first. Before you specify a rule, consider the order in which you want the data to be replicated.

You can specify the following rules:

**ALL\_DATA**

Replicates active and inactive backup data, archive data, or space-managed data. The data is replicated with a normal priority.

**ACTIVE\_DATA**

Replicates only active backup data. The data is replicated with a normal priority. This rule is valid only for **BKREPLRULEDEFAULT**.

**Attention:** If you specify **ACTIVE\_DATA** and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the **FORCERECONCILE=YES** parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

**ALL\_DATA\_HIGH\_PRIORITY**

Replicates active and inactive backup data, archive data, or space-managed data. Data is replicated with a high priority.

### **ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the ACTIVE\_DATA replication rule except data is replicated with a high priority. This rule is valid only for **BKREPLRULEDEFAULT**.

### **DEFAULT**

Replicates data according to the server replication rule for backup data.

For example, suppose that you want to replicate the archive data in all the file spaces that belongs to a client node. Replication of the archive data is a high priority. One method to accomplish this task is to specify **ARREPLRULEDEFAULT=DEFAULT**. Ensure that the file space rules for archive data are also set to **DEFAULT** and that the server rule for archive data is set to **ALL\_DATA\_HIGH\_PRIORITY**.

**Restriction:** If a node is configured for replication, the file space rules are set to **DEFAULT** after the node stores data on the source replication server.

### **NONE**

Data of the specified type is not replicated.

For example, if you do not want to replicate space-managed data that belongs to a client node, specify **SPREPLRULEDEFAULT=NONE**

### **RECOVERDAMAGED**

Specifies whether damaged files can be recovered for this node from a target replication server. The parameter is optional. The default value is **YES**. You can specify one of the following values:

#### **Yes**

Specifies that recovery of damaged files from a target replication server is enabled for this node.

**No** Specifies that recovery of damaged files from a target replication server is not enabled for this node.

**Tip:** The value of the **RECOVERDAMAGED** parameter is only one of several settings that determine whether damaged files are recovered. For information about how to specify the settings, see Settings that affect the recovery of damaged files.

### **ROLEOVERRIDE**

Specifies whether to override the reported role of the client for processor value unit (PVU) estimation reporting. The default is **USERREPORTED**. The parameter is optional.

The role reported by the client is either client-device (for example, a workstation) or server-device (for example, file/print server, application server, database). By default, the client reports its role that is based on the client type and the operating system. All clients initially report their role as server-device, except for Backup-Archive Clients running Microsoft Windows workstation distributions (Windows Vista) and Macintosh OS X.

Specify one of the following values:

#### **Client**

Specifies a client-device.

#### **Server**

Specifies a server-device.



**Other**

Specifies that this node is not to be used for PVU estimation reporting. This value can be useful when multiple nodes are deployed for a physical system (for example, virtual environments, test nodes, retired nodes, and nodes not in production or clustering).

**Usereported**

Use the reported role that is provided by the client.

**AUTHentication**

This parameter specifies the password authentication method for the node. Specify one of the following values: LDAP or LOCAL. The parameter is optional and defaults to LOCAL. The default can change to LDAP if you use the **SET DEFAULTAUTHENTICATION** command and specify LDAP.

**Local**

Specifies that the local Tivoli Storage Manager server database is used.

**LDap**

Specifies that the node uses an LDAP server for password authentication.

**Tip:** A password is not required if you register a node and select AUTHENTICATION=LDAP. At logon, you are prompted for a password.

**SSLrequired**

Specifies whether the node must use Secure Sockets Layer (SSL) to communicate with the Tivoli Storage Manager server. The parameter is optional. When you authenticate passwords with an LDAP directory server, you must protect the sessions by using SSL or another network security method.

**Yes**

Specifies that SSL is required.

**No** Specifies that SSL is not required.

**DEFault**

Specifies that SSL is required for a node if its password authenticates with an LDAP directory server. SSL is not required for a node that authenticates its password with the Tivoli Storage Manager server (LOCAL).

**SERVERonly**

Specifies that SSL is required between the backup-archive client and the server. SSL is not required between the backup-archive client and the storage agent.

**SPLITLARGEObjects**

Specifies whether large objects that are stored by this node are automatically split into smaller pieces, by the server, to optimize server processing. The parameter is optional. Specifying Yes causes the server to split large objects (over 10 GB) into smaller pieces when stored by a client node. Specifying No bypasses this process. Specify No only if your primary concern is maximizing throughput of backups directly to tape. The default value is Yes.

**Example: Register a client node that only the root user can back up**

Register the client node `mete@rite` with password `KingK0ng` to back up files from only the root user to the server.

```
register node mete@rite KingK0ng
backupinit=root
```

### **Example: Register a client node and password and set compression on**

Register the client node JOE0S2 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 JOE0S2 administrative user ID with password *SECRETCODE*. In addition, the administrator now has client owner authority to the JOE0S2 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, *HELPPADMIN*, when you register the client node JAN. This step would not automatically create an administrator ID named JAN, but would grant client owner authority for this node to the *HELPPADMIN* 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 two copy storage pools defined. The primary storage pool is of disk device type, and the two copy storage pools are of sequential device type. Specify 2 as the maximum number of mount points. The client's network was unstable during the last few weeks. You must ensure that the client data is not corrupted when it is sent over the network. Specify the **VALIDATE PROTOCOL** parameter to validate all data that is sent from the client 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
```

### Example: Register a node and allow it to deduplicate data on the client system

Register a node name of JIM and allow it to deduplicate data on the client system.

```
register node jim jim deduplication=clientorserver
```

### Example: Register a node name of ED and set the role as a server-device for PVU estimation reporting

Register a node name of ED and set the role as a server-device for PVU estimation reporting.

```
register node ed pw45twx roleoverride=server
```

### Example: Register a node on a source replication server

Define NODE1 to a source replication server. Specify a replication rule for the backup data that belongs to NODE1 so that active backup data is replicated with a high priority. Enable replication for the node.

```
register node node1 bkreplruledefault=active_data_high_priority replstate=enabled
```

### Example: Register a node that uses a password that authenticates with an LDAP server

Register a node name of node17 that must authenticate with an LDAP server.

```
register node node17 n0de17n0de#seventeen authentication=ldap
```

You must enter a new password the next time you log on if the new password does not adhere to requirements for a “complex” password.

### Example: Register a node and enable recovery of damaged files

Register a node name of PAYROLL. For the PAYROLL node, enable the recovery of damaged files from a target replication server.

```
register node payroll recoverdamaged=yes
```

## Related commands

Table 289. Commands related to REGISTER NODE

Command	Description
DEFINE ASSOCIATION	Associates clients with a schedule.
DEFINE DATAMOVER	Defines a data mover to the Tivoli Storage Manager server.
DEFINE MACHNODEASSOCIATION	Associates a Tivoli Storage Manager node with a machine.
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
LOCK NODE	Prevents a client from accessing the server.
QUERY FILESPACE	Displays information about data in file spaces that belong to a client.

Table 289. Commands related to REGISTER NODE (continued)

Command	Description
QUERY NODE	Displays partial or complete information about one or more clients.
QUERY PVUESTIMATE	Displays an estimate of the client-devices and server-devices being managed.
QUERY REPLNODE	Displays information about the replication status of a client node.
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.
REMOVE REPLNODE	Removes a node from replication.
RENAME NODE	Changes the name for a client node.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
RESET PASSEXP	Resets the password expiration for nodes or administrators.
SET DEFAULTAUTHENTICATION	Specifies the default password authentication method for any REGISTER NODE or REGISTER ADMIN commands.
SET PASSEXP	Specifies the number of days after which a password is expired and must be changed.
SET CPUINFOREFRESH	Specifies the number of days between client scans for workstation information used for PVU estimates.
SET DEDUPVERIFICATIONLEVEL	Specifies the percentage of extents verified by the server during client-side deduplication.
SET REGISTRATION	Specifies whether users can register themselves or must be registered by an administrator.
SET REPLRECOVERDAMAGED	Specifies whether node replication is enabled to recover damaged files from a target replication server.
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 FILESPACE	Changes file-space node-replication rules.
UPDATE NODE	Changes the attributes that are associated with a client node.

---

## REMOVE commands

Use the **REMOVE** commands to remove an object from Tivoli Storage Manager.

- “REMOVE ADMIN (Delete an administrator user ID)” on page 1008
- “REMOVE NODE (Delete a node or an associated machine node)” on page 1010
- “REMOVE REPLNODE (Remove a client node from replication)” on page 1012
- “REMOVE REPLSERVER (Remove a replication server)” on page 1014

## REMOVE ADMIN (Delete an administrator user ID)

Use this command to remove an administrator user ID from the system.

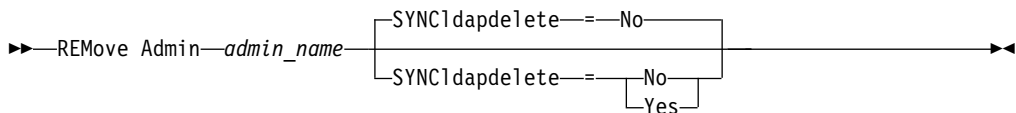
You cannot remove the last system administrator user ID or the SERVER\_CONSOLE administrative ID from the system.

Using SYNCLDAPDELETE=YES with this command removes the administrator user ID from the Lightweight Directory Access Protocol (LDAP) server. Do not specify SYNCLDAPDELETE=YES if the LDAP namespace is shared by multiple Tivoli Storage Manager servers. If you want to remove an administrator user ID from all Tivoli Storage Manager servers and all LDAP servers, specify SYNCLDAPDELETE=YES.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *admin\_name* (Required)

Specifies the name of the administrator to be removed.

#### SYNCLDAPDELETE

If the administrative user ID is stored on the LDAP server, you can delete the entry on that server by specifying a value of YES. Do not delete the entry if it is used by another server that shares the administrator namespace on the LDAP server.

### Example: Remove an administrator user ID

Remove the administrator user ID larry. Issue the following command:

```
remove admin larry
```

### Example: Remove any corresponding entries for administrator user IDs that authenticate to multiple LDAP servers

To remove the entry in the LDAP server for administrator user ID harry, issue the following command:

```
remove admin harry syncldapdelete=yes
```

### Related commands

Table 290. Commands related to REMOVE ADMIN

Command	Description
LOCK ADMIN	Prevents an administrator from accessing Tivoli Storage Manager.
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.

*Table 290. Commands related to REMOVE ADMIN (continued)*

<b>Command</b>	<b>Description</b>
REGISTER ADMIN	Defines a new administrator without granting administrative authority.
RENAME ADMIN	Changes a 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 part of a collocation group and you remove the node from the server, the node is removed from the collocation group. If a node is removed and the node contained file spaces in a file space collocation group, those file spaces are removed from the group member list.

If you remove a node that stored data in a deduplicated storage pool, the node name DELETED is displayed in the **QUERY OCCUPANCY** command output until all data deduplication dependencies are removed.

Using `SYNCLDAPDELETE = YES` with this command removes the node's entry from the LDAP directory server. Do not specify `SYNCLDAPDELETE = YES` if the LDAP namespace is shared by multiple IBM Tivoli Storage Manager servers. If you want to remove a node from all Tivoli Storage Manager servers and all LDAP directory servers, specify `SYNCLDAPDELETE = YES`.

When a node is removed, the corresponding administrative ID is removed only if the following issues are true:

- The administrator name is identical to the node name.
- The administrator has client owner or client access authority *only* to the node that is being removed.
- The administrator is not a managed object.

**Requirement:** Before you can remove a node, you must delete all backup and archive file spaces that belong to that node.

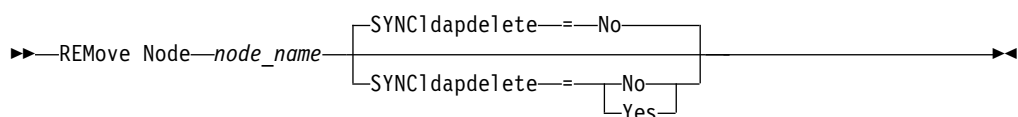
Before you can remove a NAS node that has a corresponding data mover, you must complete the following tasks in order:

1. Delete any paths from the data mover
2. Delete the data mover
3. Delete all virtual file space definitions for the node
4. 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





## Parameters

### *node\_name* (Required)

Specifies the name of the node to be removed.

### **SYNCDAPDELETE**

If the node ID is stored on the LDAP directory server, you can delete the entry on that server by specifying a value of YES. Do not delete the entry if it is used by another server that shares the namespace on the LDAP directory server.

## Example: Remove a client node

Remove the client node LARRY.

```
remove node larry
```

## Example: Remove any corresponding entries for a node ID that authenticates to an LDAP directory server

To remove the client node elroy that has corresponding entries that share namespace on an LDAP directory server, issue the following command:

```
remove node elroy syncldapdelete=yes
```

## Related commands

Table 291. 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 file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
DELETE PATH	Deletes a path from a source to a destination.
DELETE VIRTUALFSMAPPING	Delete a virtual file space mapping.
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 Tivoli Storage Manager.
REGISTER NODE	Defines a client node to the server and sets options for that user.
RENAME NODE	Changes the name for a client node.

## REMOVE REPLNODE (Remove a client node from replication)

Use this command to remove a node from replication if you no longer want to replicate the data that belongs to the node.

You cannot delete client node data by issuing the **REMOVE REPLNODE** command. You can issue the command on a source or on a target replication server. You can only issue this command from an administrative command-line client. You cannot issue this command from the server console.

If you issue the **REMOVE REPLNODE** command for a client node whose replication mode is set to **SEND** or **RECEIVE**, the mode is set to **NONE**. The replication state is also set to **NONE**. After you remove a client node from replication, the target replication server can accept backup, archive, and space-managed data directly from the node.

If a client node is removed from replication, information in the database about replication for the node is deleted. If the client node is enabled for replication later, the replication process replicates all the data that is specified by replication rules and settings.

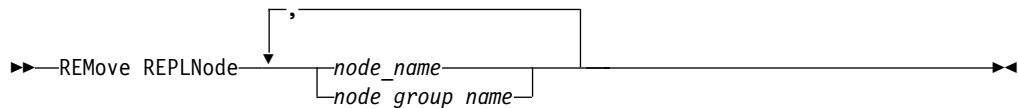
When you issue the **REMOVE REPLNODE** command, the data that belongs to a client node is not deleted. To delete file space data that belongs to the client node, issue the **DELETE FILESPACE** command for each of the file spaces that belong to the node. If you do not want to keep the client node definition, issue the **REMOVE NODE** command. To delete file space data and the client node definition, issue **DELETE FILESPACE** and **REMOVE NODE** on the target replication server.

**Restriction:** If a node replication process is running for a client node that is specified by this command, the command fails and the replication information for the node is not removed.

### Privilege class

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

### Syntax



### Parameters

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

Specifies the name of the client node or defined group of client nodes that you want to remove from replication. To specify multiple client node names and client-node group names, separate the names with commas and no intervening spaces. You can use wildcard characters to specify client node names, but not to specify client-node group names. You cannot combine node or node group names with the domain name.

## Example: Remove three client nodes and a client node group from replication

The names of the client nodes are NODE1, NODE2, and NODE3. The name of the client node group is PAYROLL. Issue the following command on the source and target replication servers:

```
remove replnode node*,payroll
```

## Related commands

*Table 292. Commands related to REMOVE REPLNODE*

Command	Description
QUERY NODE	Displays partial or complete information about one or more clients.
QUERY REPLICATION	Displays information about node replication processes.

## REMOVE REPLSERVER (Remove a replication server)

Use this command to remove or to switch to a replication server from the list of replication servers. This command deletes all information about replication state for all nodes that were replicated to that server.

You can issue the command on a source or on a target replication server.

**Restriction:** You cannot delete client node data by using the **REMOVE REPLSERVER** command.

Use the command to switch replication servers and to remove replication information for an old server. The command does not affect the current replication mode or state of any node definitions. Issue the command on both the source and target servers to keep the replication state information about both servers consistent.

**Restriction:** If you specify the default replication server for the **REMOVE REPLSERVER** command and a node replication process is running, the command fails and no replication information is removed.

This command runs as a background operation and it cannot be canceled. Tivoli Storage Manager deletes replication information that is associated with the specified server as a series of batch database transactions. If a system failure occurs, a partial deletion can occur.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—REMOve REPLServer—*GUID*—————►►

### Parameters

*replication\_guid* (Required)

The unique identifier for the replication server that is being removed. You can use wildcards to specify the Replication Global Unique Identifier (GUID), however, only one GUID can match the wildcard. If the wildcard sequence matches more than one GUID, the command fails. You must qualify the wildcard string until only the GUID that you want to delete is found.

### Example: Use a wildcard to remove a replication server

Remove a replication server by using a wildcard character to indicate the GUID.

```
remove replserver e*
```

### Related commands

Table 293. Commands related to **REMOVE REPLSERVER**

Command	Description
"REMOVE REPLNODE (Remove a client node from replication)" on page 1012	Removes a node from replication.

Table 293. Commands related to **REMOVE REPLSERVER** (continued)

Command	Description
"QUERY REPLSERVER (Query a replication server)" on page 881	Displays information about replicating servers.

---

## RENAME commands

Use the **RENAME** commands to change the name of an existing object.

- “RENAME ADMIN (Rename an administrator)” on page 1017
- “RENAME FILESPACE (Rename a client file space on the server)” on page 1019
- “RENAME NODE (Rename a node)” on page 1023
- “RENAME SCRIPT (Rename a Tivoli Storage Manager script)” on page 1025
- “RENAME SERVERGROUP (Rename a server group)” on page 1026
- “RENAME STGPOOL (Change the name of a storage pool)” on page 1027

## 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 assign an existing administrative user ID to another person, use the **UPDATE ADMIN** command to change the password.

When an administrator and a node share a name and you change the administrator authentication method, the node authentication method also changes. If you rename an administrator to the same name as an existing node, the authentication method and the **SSLREQUIRED** setting for the node can change. If those settings are different, after the renaming, both administrator and node will have the same authentication method and **SSLREQUIRED** setting.

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—————►
```

```
► [SYNCDAPDELETE=No] —————►
```

```
   |SYNCDAPDELETE=No
```

```
   |SYNCDAPDELETE=Yes
```

The diagram shows the command syntax with a long arrow pointing right. Below it, a bracketed structure shows the options for the SYNCDAPDELETE parameter, with 'No' as the default and 'Yes' as an alternative.

### Parameters

#### *current\_admin\_name* (Required)

Specifies the name of the administrator user ID to be renamed.

#### *new\_admin\_name* (Required)

Specifies the new administrator name. The maximum length of the name is 64 characters.

#### **SYNCDAPDELETE**

If the administrative user ID is stored on a Lightweight Directory Access Protocol (LDAP) server, you can rename the entry on that server by specifying a value of YES. Do not rename the entry if it is used by another server that shares the administrator namespace on the LDAP server. The default is NO.

#### **Yes**

Select YES to delete the corresponding LDAP entry in order to rename it. If you want to rename all of the same-named administrators to keep them in sync with the LDAP server, select YES.

**No** Does not delete the corresponding entries. Select NO if the corresponding entries are being used by another server that shares administrative namespace in the LDAP server. Select NO if you want to rename only one administrator and have the administrators on the shared LDAP server keep their same name. After you issue the command, issue the `UPDATE ADMIN AUTHENTICATION=LDAP` command to create an LDAP entry for the new

administrator name. Take this step to ensure that the administrator can authenticate against the LDAP server. The default is NO.

### **Example: Rename an administrator**

Rename the Tivoli Storage Manager administrator CLAUDIA to BILL.

```
rename admin claudia bill
```

### **Example: Rename an administrator and all same-named administrators that authenticate to an LDAP server**

```
rename admin claudia bill syncldapdelete=yes
```

### **Related commands**

*Table 294. Commands related to RENAME ADMIN*

<b>Command</b>	<b>Description</b>
QUERY ADMIN	Displays information about one or more 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.

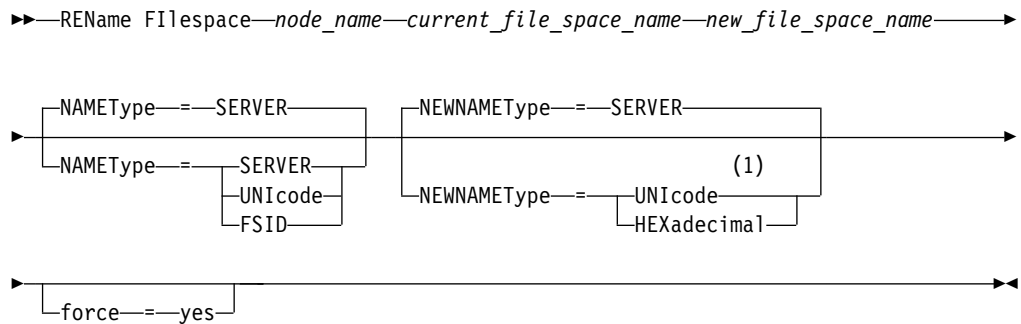
You might want to rename a file space that was imported or to cause the creation of new Unicode-enabled file spaces for Unicode-enabled clients.

**Restriction:** Do not rename NAS or VMware file spaces. If you rename a NAS or VMware file space, it is no longer visible and cannot be restored. To restore a renamed NAS or VMware file space, you must rename it back to its original name and set the force parameter as follows: **force=yes**

### Privilege class

Any administrator with unrestricted policy authority or with restricted policy authority over the client's policy domain can issue this command.

### Syntax



### 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.

**Important:** If the new name type is hexadecimal, 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.

When you rename a file space that is part of a file space collocation group, the collocation group is updated with the new name.

#### **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 with Windows, Macintosh OS X, and NetWare operating systems.

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 that is 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 cannot access 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 that is 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 might want to specify the HEXADECEIMAL 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, and use a new name in the server's code page. You cannot mix the two types.

#### *force*

To rename a NAS or VMware file space you must set this parameter as follows: **force=yes**

## 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 reimported to the server. When this file space was imported, a system-generated name, /r031, was created for it because /r033 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 overlays 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
```

## Rename file space to create a Unicode-enabled file space

Client JOE is using an English Unicode-enabled Tivoli Storage Manager client. JOE backed up several large file spaces that are not Unicode that is 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 that is 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 now. The administrator wants to rename only 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 allows 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 295. Commands related to RENAME FILESPACE*

Command	Description
DEFINE VIRTUALFSMAPPING	Define a virtual file space mapping.
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
EXPORT NODE	Copies client node information to external media or directly to another server.
QUERY FILESPACE	Displays information about data in file spaces that belong to a client.

*Table 295. Commands related to RENAME FILESPACE (continued)*

<b>Command</b>	<b>Description</b>
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, use the **UPDATE NODE** command to change the password.

### Restrictions:

- 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.
- If a node is configured for replication, it cannot be renamed.

If you rename a node to the same name as an existing administrator, the administrator authentication method and **SSLREQUIRED** setting are updated to match the node. When a node and an administrator share a name and you change the node authentication method or the node **SSLREQUIRED** setting, the administrator settings also change. You must have system level authority to update the node authentication method or the node **SSLREQUIRED** setting and also update a same-named administrator.

### Privilege class

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—————►
```

```
┌──SYNCldapdelete—=—No──┐  
└──SYNCldapdelete—=—┬──No──┐  
                     └──Yes──┘
```

### 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 is 64 characters.

#### **SYNCldapdelete**

If the node ID is stored on a Lightweight Directory Access Protocol (LDAP) server, you can rename the entry on that server by specifying a value of YES. Do not rename the entry if it is used by another server that shares the namespace on the LDAP directory server. This parameter applies to nodes that authenticate passwords with an LDAP directory server. The default is NO.

#### **Yes**

Select YES to delete the corresponding LDAP entry in order to rename it. If you want to rename all of the same-named nodes to keep them in sync with the LDAP server, select YES.

**No** Does not delete the corresponding entries. Select NO if the corresponding entries are being used by another server that shares administrative namespace in the LDAP directory server. Select NO if you want to rename

only one node and keep the same-named nodes on the shared LDAP server, the same. After you issue the command, issue the UPDATE NODE AUTHENTICATION=LDAP command to create an LDAP entry for the new node name. Updating the node this way ensures that the node can authenticate against the LDAP directory server. The default is NO.

### Example: Rename a node

Rename the node JOE to JOYCE.

```
rename node joe joyce
```

### Example: Rename a node that shares a namespace with other servers

Rename the node JOYCE to JOE and do not delete the previous name from corresponding LDAP directory servers. Specify **SYNCLDAPDELETE=NO** to ensure that other servers that share a namespace on the LDAP directory server can still connect to the LDAP directory server.

```
rename node joyce joe syncldapdelete=no
```

### Related commands

*Table 296. Commands related to RENAME NODE*

Command	Description
QUERY NODE	Displays partial or complete information about one or more clients.
UPDATE NODE	Changes the attributes that are 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 297. Commands related to RENAME SCRIPT

Command	Description
COPY SCRIPT	Creates a copy of a script.
DEFINE SCRIPT	Defines a script to the 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 298. 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 to use the same names on a configuration manager and its managed servers.

When you rename a storage pool, any administrators with restricted storage privilege for the old storage pool automatically retain restricted storage privilege for the renamed storage pool. If the renamed storage pool is in a storage pool hierarchy, the hierarchy is preserved. You must update the management class or copy group to specify the new storage pool name as the destination for files.

If processes are active when a storage pool is renamed, the old name might 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 299. 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.

## REPLICATE NODE (Replicate data in file spaces that belong to a client node)

Use this command to replicate data in file spaces that belong to one or more client nodes or defined groups of client nodes.

When you issue this command, a process is started in which data that belongs to the specified client nodes is replicated according to replication rules. Files that are no longer stored on the source replication server, but that exist on the target replication server, are deleted during this process.

If a node replication process is already running for a client node that is specified by this command, the node is skipped, and replication begins for other nodes that are enabled for replication.

After the node replication process is completed, a recovery process can be started on the target replication server. Files are recovered only if all the following conditions are met:

- Tivoli Storage Manager, Version 7.1.1 or later, is installed on the source and target replication servers.
- The **REPLRECOVERDAMAGED** system parameter is set to ON. The system parameter can be set by using the **SET REPLRECOVERDAMAGED** command.
- The source server includes at least one file that is marked as damaged in the node that is being replicated.
- The node data was replicated before the damage occurred.

The following table describes how settings affect the recovery of damaged, replicated files.

**Restriction:** You cannot use the **REPLRECOVERDAMAGED** parameter for directory-container or cloud storage pools.

*Table 300. Settings that affect the recovery of damaged files.*

Setting for the <b>REPLRECOVERDAMAGED</b> system parameter	Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REPLICATE NODE</b> command	Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REGISTER NODE</b> and <b>UPDATE NODE</b> commands	Result
OFF	YES, NO, or not specified	YES or NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.
OFF	ONLY	YES or NO	An error message is displayed because files cannot be recovered when the <b>REPLRECOVERDAMAGED</b> system parameter is set to OFF.

Table 300. Settings that affect the recovery of damaged files (continued).

Setting for the REPLRECOVERDAMAGED system parameter	Value of the RECOVERDAMAGED parameter on the REPLICATE NODE command	Value of the RECOVERDAMAGED parameter on the REGISTER NODE and UPDATE NODE commands	Result
ON	YES	YES or NO	During node replication, standard replication occurs and damaged files are recovered from the target replication server.
ON	NO	YES or NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.
ON	ONLY	YES or NO	Damaged files are recovered from the target replication server, but standard node replication does not occur.
ON	Not specified	YES	During node replication, standard replication occurs and damaged files are recovered from the target replication server.
ON	Not specified	NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.

**Tip:** When the **QUERY PROCESS** command is issued during node replication, the output can show unexpected results for the number of completed replications. The reason is that, for node replication purposes, each file space is considered to contain three logical file spaces:

- One for backup objects
- One for archive objects
- One for space-managed objects

By default, the **QUERY PROCESS** command generates results for each logical file space. Other factors also affect the output of the **QUERY PROCESS** command:

- If a file space has a replication rule that is set to **NONE**, the file space is not included in the count of file spaces that are being processed.

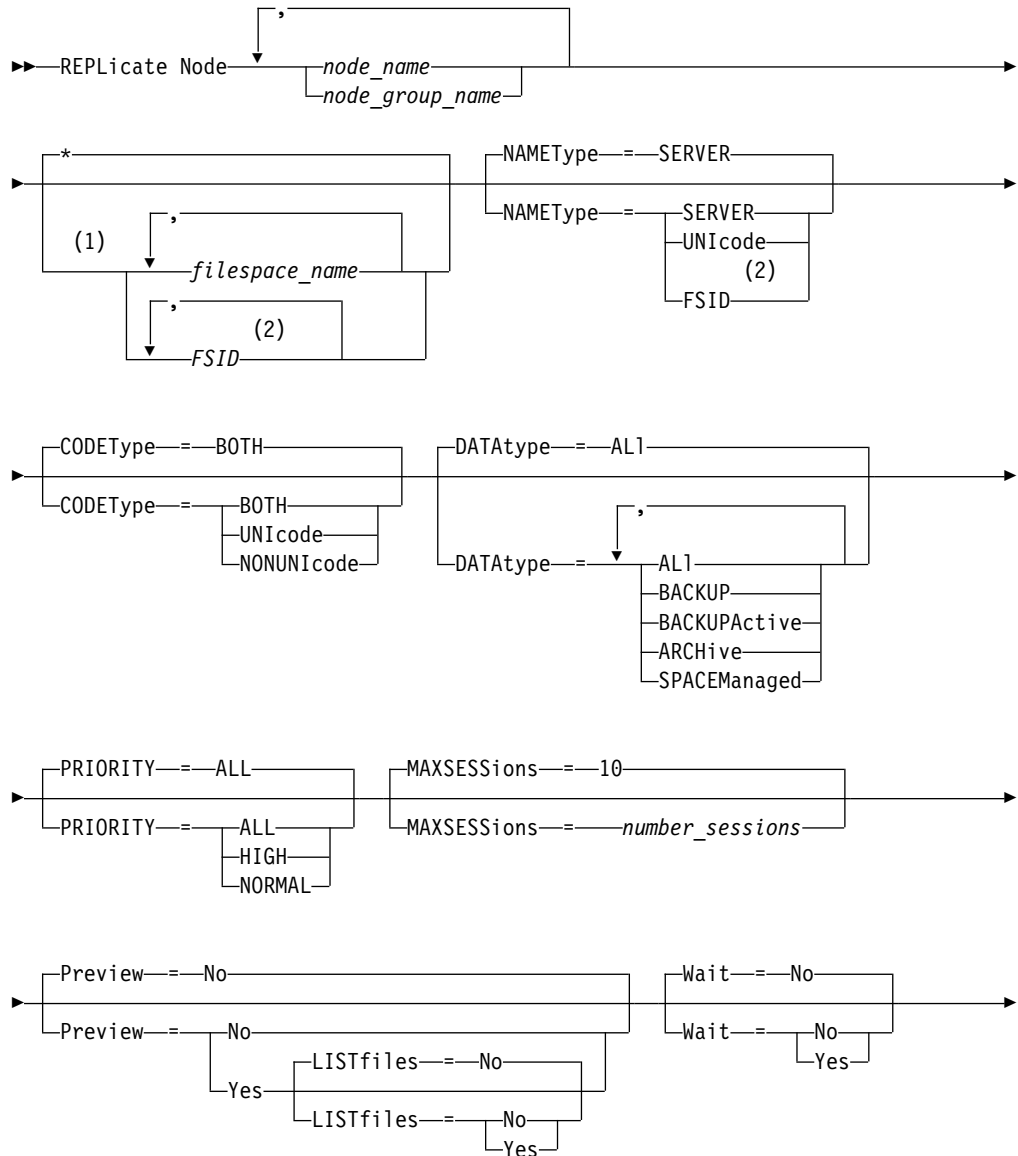
- If you specify data types in the **REPLICATE NODE** command, only those data types are included in the count of file spaces that are being processed, minus any file spaces that are excluded.

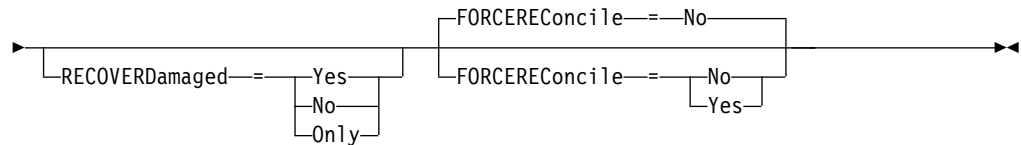
Issue this command on the server that acts as a source for replicated data.

## Privilege class

To issue this command, you must have system privilege.

## Syntax





**Notes:**

- 1 Do not mix file space identifiers (FSIDs) and file space names in the same command.
- 2 Do not specify FSID if you use wildcard characters for the client node name.

**Parameters**

*node\_name* or *node\_group\_name* **(Required)**

Specifies the name of the client node or defined group of client nodes whose data is to be replicated. You can also specify a combination of client node names and client-node group names. To specify multiple client node names or client-node group names, separate the names with commas with no intervening spaces. You can use wildcard characters with client node names but not with client-node group names. The replication rules for all file spaces in the specified client nodes are checked.

*filespace\_name* or *FSID*

Specifies the name of the file space or the file space identifier (FSID) to be replicated. A name or FSID is optional. If you do not specify a name or an FSID, all the data in all the file spaces for the specified client nodes is eligible for replication.

*filespace\_name*

Specifies the name of the file space that has data to be replicated. File space names are case-sensitive. To determine the correct capitalization for the file space, issue the **QUERY FILESPACE** command. Separate multiple names with commas with no intervening spaces. When you specify a name, you can use wildcard characters.

A server that has clients with file spaces that are enabled for Unicode might have to convert the file space name. For example, the server might have to convert a name from the server code page to Unicode. For details, see the **NAMETYPE** parameter. If you do not specify a file space name, or if you specify 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.

*FSID*

Specifies the file space identifier for the file space to be replicated. The server uses FSIDs to find the file spaces to replicate. To determine the FSID for a file space, issue the **QUERY FILESPACE** command. Separate multiple FSIDs with commas with no intervening spaces. If you specify an FSID, the value of the **NAMETYPE** parameter must be FSID.

**NAMETYPE**

Specifies how you want the server to interpret the file space names that you enter. You can use this parameter for Tivoli Storage Manager clients that are enabled for Unicode and that have Windows, Macintosh OS X, or NetWare operating systems.

Use this parameter only when you enter a partly qualified or fully qualified file space name. The default value is SERVER. You can specify one of the following values:

**SERVER**

The server uses the server code page to interpret file space names.

**UNICODE**

The server converts file space names from the server code page to the UTF-8 code page. The success of the conversion depends on the characters in the name and the server code page. Conversion can fail if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines.

**FSID**

The server interprets file space names by using their file space identifiers.

**CODETYPE**

Specifies the type of file spaces to be included in node replication processing. Use this parameter only when you enter a single wildcard character for the file space name. The default value is BOTH, which specifies that file spaces are included regardless of code page type. You can specify one of the following values:

**UNICODE**

Specifies file spaces that are only in Unicode.

**NONUNICODE**

Specifies file spaces that are not in Unicode.

**BOTH**

Specifies all file spaces regardless of code page type.

**DATATYPE**

Specifies the type of data to be replicated. Data is replicated according to the replication rule that applies to the data type. This parameter is optional. You can specify one or more data types. If you do not specify a data type, all backup, archive, and space-managed data is replicated. Separate multiple data types with commas with no intervening spaces. You cannot use wildcard characters. You can specify one of the following values:

**ALL**

Replicates all backup, archive, and space-managed data in a file space according to the rule that is assigned to the data type. For example, suppose that NODE1 has a single file space. The following replication rules apply:

- The file space rules for backup and archive data in the file space are set to ALL\_DATA.
- The file space rule for space-managed data is set to DEFAULT.
- The client node rule for space-managed data is set to NONE.

If you issue REPLICATE NODE NODE1 DATATYPE=ALL, only backup data and archive data are replicated.

**BACKUP**

Replicates active and inactive backup data in a file space if the controlling replication rule is ALL\_DATA, ACTIVE\_DATA, ALL\_DATA\_HIGH\_PRIORITY, or ACTIVE\_DATA\_HIGH\_PRIORITY.

**BACKUPActive**

Replicates only active backup data in a file space if the controlling replication rule is `ACTIVE_DATA` or `ACTIVE_DATA_HIGH_PRIORITY`.

**ARCHive**

Replicates archive data only in a file space if the controlling replication rule is `ALL_DATA` or `ALL_DATA_HIGH_PRIORITY`.

**SPACEManaged**

Replicates only space-managed data in a file space if the controlling replication rule is `ALL_DATA` or `ALL_DATA_HIGH_PRIORITY`.

**PRIority**

Specifies the data to replicate based on the priority of the replication rule. You can specify one of the following values:

**All**

Replicates all data in a file space if the controlling replication rule is `ALL_DATA`, `ACTIVE_DATA`, `ALL_DATA_HIGH_PRIORITY`, or `ACTIVE_DATA_HIGH_PRIORITY`.

**High**

Replicates only data in a file space that has a controlling replication rule of `ALL_DATA_HIGH_PRIORITY` or `ACTIVE_DATA_HIGH_PRIORITY`.

**Normal**

Replicates only data in a file space that has a controlling replication rule of `ALL_DATA` or `ACTIVE_DATA`.

**MAXSESSions**

Specifies the maximum allowable number of data sessions to use for sending data to a target replication server. This parameter is optional. The value can be 1 - 99. The default value is 10.

Increasing the number of sessions can improve node replication throughput.

When you set this value, consider the number of logical and physical drives that can be dedicated to the replication process. 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 the following factors:

- Other Tivoli Storage Manager and system activity
- The mount limits of the device classes for the sequential access storage pools that are involved

Ensure that sufficient mount points and drives are available to allow node replication processes to complete. Each replication session might need a mount point on the source and target replication servers for storage pool volumes. If the device type is not `FILE`, each session might also need a drive on both the source and target replication servers.

When you set a value for **MAXSESSIONS**, also consider the available bandwidth and the processor capacity of the source and target replication servers.

**Tip:**

- The value that is specified by the **MAXSESSIONS** parameter applies only to data sessions. Data sessions are sessions during which data is sent to a target replication server. However, if you issue a **QUERY SESSION** command, the total number of sessions might exceed the number of data sessions. The difference is because of short control sessions that are used for querying and setting up replication operations.

- The value of the **MAXSESSIONS** parameter represents the maximum allowable number of sessions. The number of sessions that are used for replication depends on the amount of data to be replicated. If you are replicating only a small amount of data, you do not achieve any benefit by increasing the number of sessions. The total number of sessions might be less than the value that is specified by the **MAXSESSIONS** parameter.

### Preview

Specifies whether to preview data. This parameter is optional. The default value is NO. You can specify one of the following values:

**No** Specifies that the data is replicated to the target server but that the data is not previewed.

### Yes

Specifies that data is previewed but not replicated. If you specify **PREVIEW=YES**, only volumes that must be physically mounted, such as tape volumes, are displayed. Volumes that are assigned to storage pools that have a device class of FILE are not displayed.

The following information is displayed in the output:

- The names of client nodes whose data would be replicated.
- The number of files that would be replicated or deleted.
- The estimated amount of time it would take to complete the node replication process.
- A list of volumes that would be mounted.
- A summary of information about replicated, damaged data. The summary lists the number of nodes, file spaces, files, and bytes that can be recovered during a replication recovery process. The summary is displayed only if **RECOVERDAMAGED=YES** or **RECOVERDAMAGED=ONLY** is specified.

If the client node data that is specified by the **REPLICATE NODE** command was never replicated and you specify **PREVIEW=YES**, the node and its file spaces are automatically defined on the target replication server.

### LISTfiles

Specifies whether to list the names of files that would be replicated. This parameter is optional. The default is NO. Specifying this parameter signifies that the **WAIT** parameter is set to YES and that you cannot issue the **WAIT** parameter from the server console.

You can specify one of the following values:

**No** Specifies that the names of files that would be replicated are not displayed.

### Yes

Specifies that the names of files that would be replicated are displayed.

### 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. You can specify one of the following values:

**No** Specifies that the command processes in the background. To monitor the background processing of the **REPLICATE NODE** command, issue the **QUERY PROCESS** command.



**Yes**

Specifies that the command processes in the foreground. Messages are not displayed until the command completes processing. You cannot specify `WAIT=YES` from the server console.

**RECOVERDamaged**

Specifies whether a recovery process is started on a target replication server after the node replication process is completed. This parameter is optional, and it overrides any value that you specified for the **RECOVERDamaged** parameter when you defined or updated a node. You can specify one of the following values:

**Yes**

Specifies that a replication process is started to recover damaged files, but only if the setting for the **REPLRECOVERDAMAGED** system parameter is ON. If the setting is OFF, damaged files are not recovered.

**No** Specifies that damaged files are not recovered.

**Only**

Specifies that a replication process is started for the sole purpose of recovering damaged files, but only if the setting for the **REPLRECOVERDAMAGED** system parameter is ON. If the setting is OFF, damaged files are not recovered, and you receive a notification that recovery was not started.

**Restriction:** If you specify an invalid combination of values and settings for file recovery, replication is stopped, and an error message is displayed.

**FORCEREconcile**

Specifies whether to compare all files on the source replication server with files on the target replication server and to synchronize the differences between them. Before Tivoli Storage Manager V7.1.1, this behavior was the default for replication processing. When Tivoli Storage Manager, V7.1.1 or later, is installed on the source and target replication servers, a reconcile is automatically completed during initial replication. After initial replication, you might use this parameter for the following reasons:

- To synchronize files on the source and target replication servers if they are different.
- To replicate inactive files that were skipped after you change your replication rules from `ACTIVE_DATA` to `ALL_DATA`.
- To delete inactive files from the target replication server when you change your replication rules from `ALL_DATA` to `ACTIVE_DATA`.
- To ensure that you replicate only active data when you are using the `ACTIVE_DATA` replication rule so that the target replication server has active files only.
- To resynchronize the files so that the target replication server has the same files as the source replication server if you have previously or are currently using the policies on the target replication server to manage replicated files.
- To resynchronize the files on the source and target replication servers if the database is regressed to an earlier point-in-time by using a method other than the **DSMSERV RESTORE DB** command.
- To rebind files to the new management class on the target replication server if this management class did not exist when the files were replicated. You must be using the policies that are defined on the target replication server to manage replicated files.

**Remember:** When the ACTIVE\_DATA rule is assigned, a reconcile is completed only for active files on the source replication server.

This parameter is optional. You can specify one of the following values:

**No** Specifies that replication processing does not force a reconcile to compare all files on the source replication server with files on the target replication server. Instead, replication processing tracks file changes on the source replication server since the last replication and synchronizes these changes on the target replication server. NO is the default value.

**Yes**

Specifies that replication processing forces a reconcile to compare all files on the source replication server with files on the target replication server and synchronizes the files on the target replication server with the source replication server.

### **Example: Replicate data by data type and priority**

Replicate high-priority active backup data and high-priority archive data that belongs to all the client nodes in group PAYROLL.

```
replicate node payroll datatype=backupactive,archive priority=high
```

### **Example: Replicate all the data that belongs to a node according to the assigned replication rules**

NODE1 has a single file space. The following replication rules apply:

- File space rules:
  - Backup data: ACTIVE\_DATA
  - Archive data: DEFAULT
  - Space-managed data: DEFAULT
- Client node rules:
  - Backup data: DEFAULT
  - Archive data: ALL\_DATA\_HIGH\_PRIORITY
  - Space-managed data: DEFAULT
- Server rules:
  - Backup data: ALL\_DATA
  - Archive data: ALL\_DATA
  - Space-managed data: NONE

```
replicate node node1 priority=all
```

Active backup data in the file space is replicated with normal priority. Archive data is replicated with high priority. Space-managed data is not replicated.

### **Example: Recover damaged files without starting the full replication process**

Without starting the full replication process, recover any damaged files in the client nodes of the PAYROLL group. Ensure that the setting for the **REPLRECOVERDAMAGED** system parameter is ON. Then, issue the following command:

```
replicate node payroll recoverdamaged=only
```

## Related commands

Table 301. Commands related to REPLICATE NODE

Command	Description
CANCEL PROCESS	Cancels a background server process.
CANCEL REPLICATION	Cancels node replication processes.
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 REPLICATION	Displays information about node replication processes.
QUERY REPLNODE	Displays information about the replication status of a client node.
QUERY REPLRULE	Displays information about node replication rules.
QUERY SERVER	Displays information about servers.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REGISTER NODE	Defines a client node to the server and sets options for that user.
REMOVE REPLNODE	Removes a node from replication.
PROTECT FILESPACE	Protects a directory-container storage pool.
SET REPLRECOVERDAMAGED	Specifies whether node replication is enabled to recover damaged files from a target replication server.
UPDATE FILESPACE	Changes file-space node-replication rules.
UPDATE NODE	Changes the attributes that are associated with a client node.
UPDATE REPLRULE	Enables or disables replication rules.
VALIDATE REPLICATION	Verifies replication for file spaces and data types.

---

## 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 302. 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 the **RESET PASSEXP** command to reset the password expiration period to the common expiration period for administrator and client node passwords. The **RESET PASSEXP** command does not apply to passwords that are stored on an LDAP directory server.

**Restriction:** You cannot reset the password expiration period to the common expiration period with the **SET PASSEXP** command.

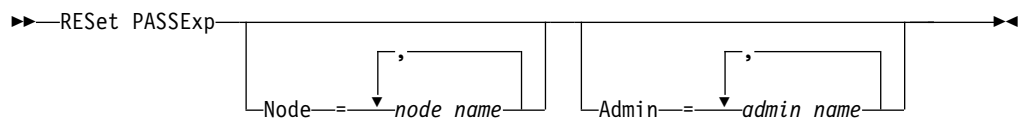
Use the **QUERY STATUS** command to display the common password expiration period.

**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 303. Commands related to RESET PASSEXP*

<b>Command</b>	<b>Description</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
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 that are associated with a client node.

---

## RESTART EXPORT (Restart a suspended export operation)

Use this command to restart a suspended export operation.

An export operation is suspended when any of the following conditions is detected:

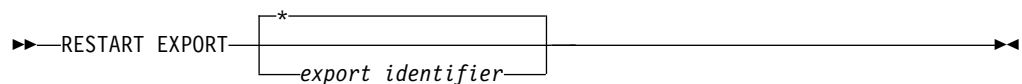
- A **SUSPEND EXPORT** command is issued for the running export operation
- Segment preemption - the file being read for export is deleted by some other process
- Communication errors on a server-to-server export
- No available mount points
- Necessary volumes are unavailable
- I/O errors encountered

**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.

### Privilege class

You must have system privilege to issue this command.

### Syntax



### Parameters

#### *export\_identifier*

This optional parameter is the unique identifier for the suspended server-to-server export operation. You can use the wildcard character to specify this name. The export identifier name can be found by issuing the **QUERY EXPORT** command to list all the currently suspended server-to-server export operations.

### Example: Restart a suspended export

Restart the suspended export operation identified by the export identifier EXPORTALLACCTNODES.

```
restart export exportallacctnodes
```

### Related commands

Table 304. Commands related to **RESTART EXPORT**

Command	Description
CANCEL EXPORT	Deletes a suspended export operation.
EXPORT NODE	Copies client node information to external media or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.

*Table 304. Commands related to RESTART EXPORT (continued)*

<b>Command</b>	<b>Description</b>
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.

- “RESTORE NODE (Restore a NAS node)” on page 1044
- “RESTORE STGPOOL (Restore storage pool data from a copy pool or an active-data pool)” on page 1049
- “RESTORE VOLUME (Restore primary volume data from a copy pool or an active-data pool)” on page 1053

## RESTORE NODE (Restore a NAS node)

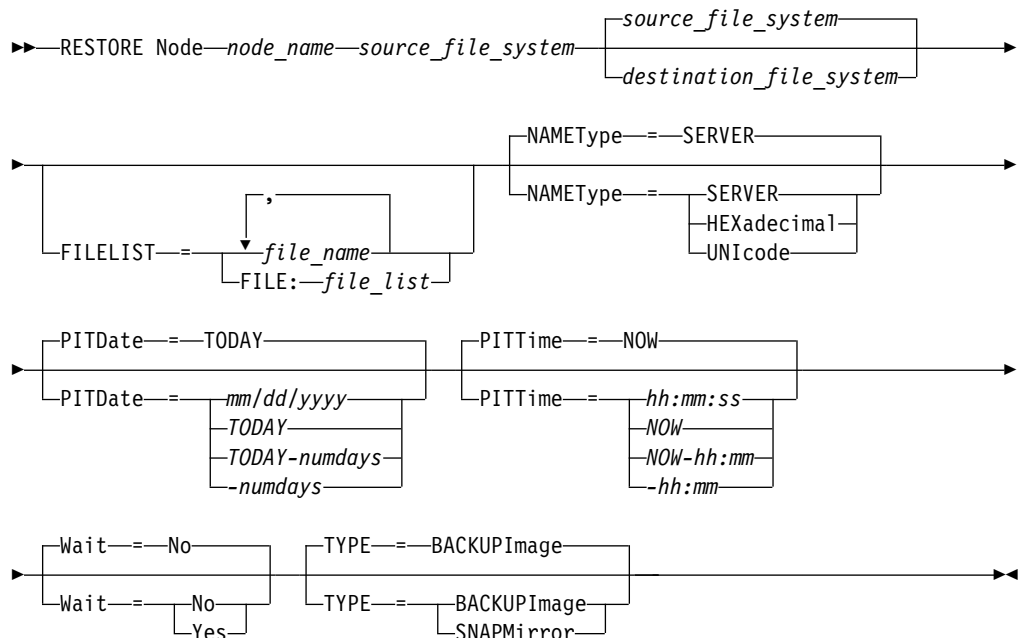
Use this command to initiate a restore operation for a network-attached storage (NAS) node.

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 is not accessed, so the 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. 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
```

To restore a complete directory, specify a directory name instead of a file name. All files in the directory and its subdirectories are restored. An ending forward slash (/) is not needed at the end of the directory name:

```
FILELIST=/path/to/mydir
```

### *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 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
<i>MM/DD/YYYY</i>	A specific date	06/25/2001
<b>TODAY</b>	The current date	<b>TODAY</b>
<i>TODAY-days</i> or <i>-days</i>	The current date minus days specified	<b>TODAY-7</b> or <b>-7</b> .  To restore data that was backed up a week ago, specify <b>PITDATE=TODAY-7</b> or <b>PITDATE=-7</b> .
<b>EOLM</b> (End Of Last Month)	The last day of the previous month.	<b>EOLM</b>
<i>EOLM-days</i>	The last day of the previous month minus days specified.	<b>EOLM-1</b>  To include files that were active a day before the last day of the previous month.
<b>BOTM</b> (Beginning Of This Month)	The first day of the current month.	<b>BOTM</b>
<i>BOTM+days</i>	The first day of the current month, plus days specified.	<b>BOTM+9</b>  To include files that were active on the 10th day of the current month.

## 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
<i>HH:MM:SS</i>	A specific time on the specified date	12:33:28
NOW	The current time on the specified date	NOW
NOW- <i>HH:MM</i> or - <i>HH:MM</i>	The current time minus hours and minutes on the specified begin date	NOW-03:30 or -03:30.  If you issue this command at 9:00 with PITTIME=NOW-03:30 or PITTIME=-03:30, the server restores backup records with a time of 5:30 or 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 filesystem name.
- This parameter is valid for NetApp and IBM N-Series file servers only.

### Example: Restore a complete directory

Restore all of the files and subdirectories in the directory /mydir.

```
restore node nasnode /myfs /dest filelist=/path/to/mydir
```

### 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
```

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 305. Commands related to RESTORE NODE

Command	Description
BACKUP NODE	Backs up a network-attached storage (NAS) node.
CANCEL PROCESS	Cancels a background server process.
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

**Restriction:** You cannot use this command for container storage pools. Use the **REPLICATE STGPOOL** command to protect data for container storage pools.

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.

**Tip:** 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:

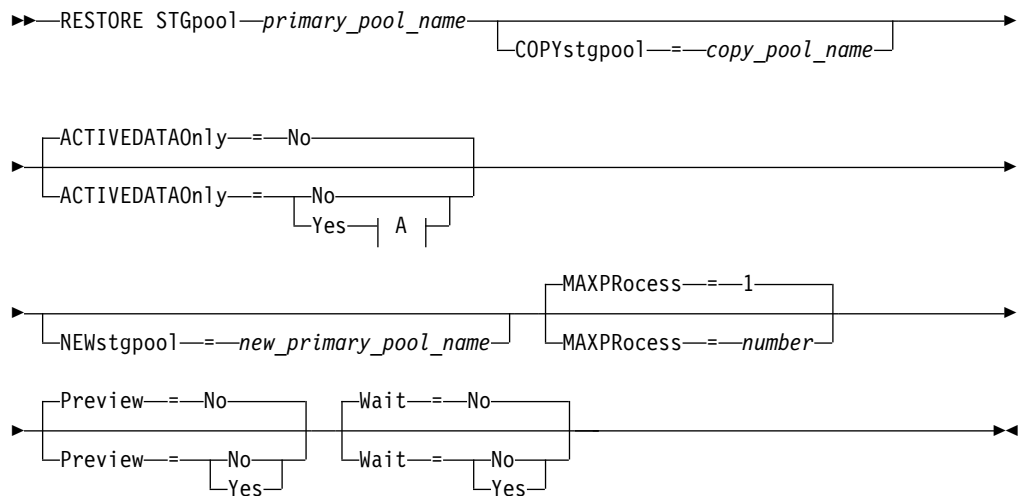
- **MOVE DATA**
- **DELETE VOLUME (DISCARDATA=YES)**
- **AUDIT VOLUME (FIX=YES)**

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



### 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 `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 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:

**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 306. 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.

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)

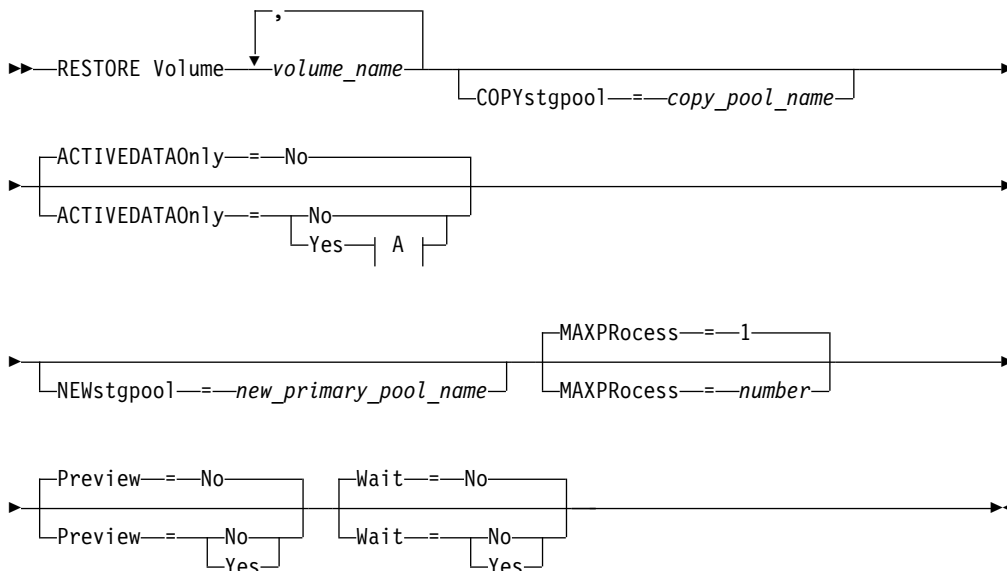
To prevent reclamation processing of copy storage pools, issue the **UPDATE STGPOOL** command with the **RECLAIM** parameter set to 100.

### 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

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 change the access mode of the identified 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 pv012`

**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 307. Commands related to RESTORE VOLUME*

Command	Description
BACKUP STGPOOL	Backs up a primary storage pool to a copy storage pool.
COPY ACTIVE DATA	Copies active backup data.
CANCEL PROCESS	Cancel a background server process.
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** commands to revoke privileges or access.

- “REVOKE AUTHORITY (Remove administrator authority)” on page 1058
- “REVOKE PROXYNODE (Revoke proxy authority for a client node)” on page 1062

## 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.

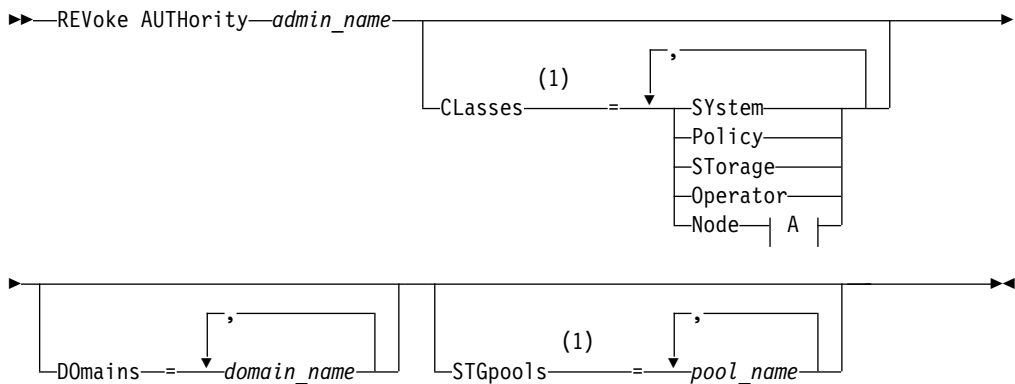
If you use the **REVOKE AUTHORITY** command without the **CLASSES**, **DOMAINS**, and **STGPOOLS** parameters, you will revoke all privileges for the specified administrator.

At least one administrator must have system privilege; therefore, if the administrator is the only one with system privilege, you cannot revoke the authority.

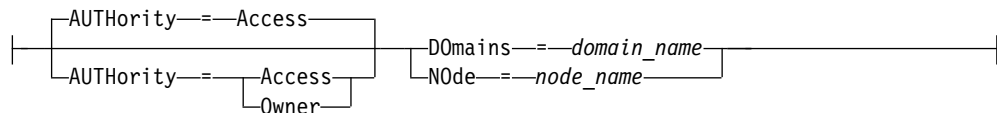
### Privilege class

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.



**SyStem**

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.

**Policy**

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.

**STorage**

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.

**Operator**

Indicates that operator privilege is to be revoked for this administrator.

**Node**

Indicates that node privilege is to be revoked for this user.

**AUTHority**

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.

**DOmains**

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.

**NOde**

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.

**DOmains**

When used with CLASSES=POLICY, specifies a list of policy domains that can no longer be managed by a restricted policy administrator. (The administrator was 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 is 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
```

## Related commands

*Table 308. Commands related to REVOKE AUTHORITY*

<b>Command</b>	<b>Description</b>
GRANT AUTHORITY	Assigns privilege classes to an administrator.
QUERY ADMIN	Displays information about one or more 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 309. 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. The macro contents are:

```
/* Macro to register policy
administrators and grant authority */
REGister Admin sara hobby
GRant AUTHority sara CClasses=Policy
REGister Admin ken plane
GRant AUTHority ken CClasses=Policy
ROLLBACK /* prevents any changes from being committed */
```

### Related commands

Table 310. 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.

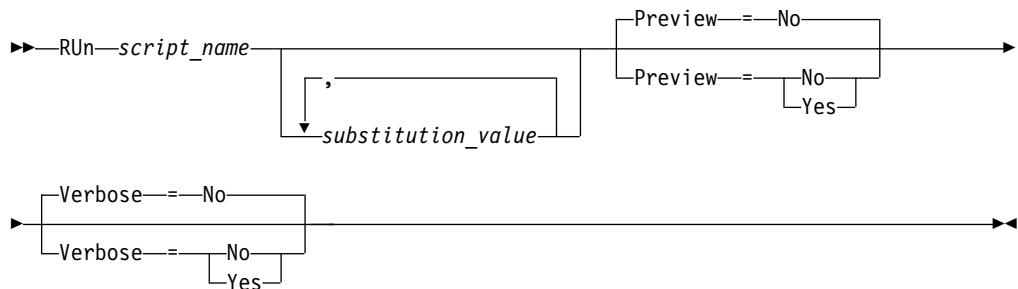
You can include **RUN** commands in scripts as long as they do not create loops. For example, you should avoid including **RUN** commands where SCRIPT\_A runs SCRIPT\_B and SCRIPT\_B runs SCRIPT\_A.

**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.

#### 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 tabname='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 311. Commands related to RUN

Command	Description
COPY SCRIPT	Creates a copy of a script.
DEFINE SCRIPT	Defines a script to the 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 Tivoli Storage Manager database)

Use the **SELECT** command to create and format a customized query of the Tivoli Storage Manager database.

Tivoli Storage Manager provides an 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 that 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 table spaces are used to process SQL queries within DB2. Inadequate temporary space can cause SQL queries to fail.

To export output to a comma-separated file for import into a spreadsheet, use `-comma` and `>` command-line options on the **dsmdmc** command.

## **Privilege class**

Any administrator can issue this command.

## **Syntax**

For **SELECT** statement syntax and guidelines, search the DB2 product information.

**Important:** The appropriate syntax for the timestamp Select statement is:

```
SELECT * FROM SUMMARY WHERE ACTIVITY='EXPIRATION' AND START_TIME  
>'2009-05-10 00:00:00' AND START_TIME <'2009-05-11 23:23:23'
```

## List of examples

The **SELECT** command is used to 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 administrator user ID passwords that are authenticated with an external LDAP directory server
- 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 is 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 off-site volumes that have their space reclaimed during reclamation of a storage pool
- Obtain PVU estimate detail records
- Obtain information about the node roles
- Obtain information about status

### **Example: List administrator user IDs that authenticate to the Tivoli Storage Manager server**

List all the administrator user IDs whose passwords authenticate with the Tivoli Storage Manager server:

```
select admin_name from admins where
authentication=local
```

### Example: List available tables

List all the tables available for querying the IBM Tivoli Storage Manager database.

```
select * from syscat.tables
```

```
ABSHEMA: SERVER1
TABNAME: ACTLOG
CREATE_TIME: 1999-05-01 07:39:06
COLCOUNT: 10
INDEX_COLCOUNT: 1
UNIQUE_INDEX: FALSE
REMARKS: Server activity log

TABSHEMA: SERVER1
TABNAME: ADMIN_SCHEDULES
CREATE_TIME: 1995-05-01 07:39:06
COLCOUNT: 14
INDEX_COLCOUNT: 1
UNIQUE_INDEX: TRUE
REMARKS: Administrative command schedules

TABSHEMA: SERVER1
TABNAME: ADMINS
CREATE_TIME: 1995-05-01 07:39:06
COLCOUNT: 15
INDEX_COLCOUNT: 1
UNIQUE_INDEX: TRUE
REMARKS: Server administrators

TABSHEMA: 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 who 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 five 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 might 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 is 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 takes 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
```

**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: 0.00000000000000
```

**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 off-site volumes that 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
MEDIAS: 0
PROCESSES: 2
SUCCESSFUL: YES
VOLUME_NAME:
DRIVE_NAME:
LIBRARY_NAME:
LAST_USE:
COMM_WAIT:
NUM_OFFSITE_VOLS: 2
```

### Example: Identify which storage pools contain data that was deduplicated by clients

```
select stgpool_name,has_client_dedup_data from stgpools
```

STGPOOL_NAME	HAS_CLIENT_DEDUP_DATA
ADPOOL	NO
ARCHIVEPOOL	NO
BACKUPPOOL	NO
COPYDEDUP	NO
COPYNODEDUP	NO
FILEPOOL	YES
FILEPOOL2	NO
LANFREEFILEPOOL	YES
SPACEMGPPOOL	NO

### Example: Obtain information about the database

```
select * from db
```

```
DATABASE_NAME: TSMDB1
TOT_FILE_SYSTEM_MB: 2048000
USED_DB_SPACE_MB: 12576
FREE_SPACE_MB: 1576871
TOTAL_PAGES: 983044
USABLE_PAGES: 982908
USED_PAGES: 977736
FREE_PAGES: 5172
BUFF_HIT_RATIO: 96.2
TOTAL_BUFF_REQ: 53967
SORT_OVERFLOW: 0
LOCK_ESCALATION: 0
PKG_HIT_RATIO: 70.0
LAST_REORG: 2010-07-15 17:32:55.000000
FULL_DEV_CLASS: OUTFILE
NUM_BACKUP_INCR: 0
LAST_BACKUP_DATE: 2010-01-21 10:37:59.000000
PHYSICAL_VOLUMES: 0
PAGE_SIZE:
NUM_BACKUP_STREAMS: 4
```

### Example: Obtain PVU estimate detail records

Generate the PVU estimate for a node named ACCTSRECSRV, which is used by the IBM Tivoli Storage Manager Extended Edition product.

```
select * from pvuestimate_details where node_name='ACCTSRECSRV'
```

```

PRODUCT: PRODTSMEE
LICENSE_NAME: MGSYSLAN
NODE_NAME: ACCTSRECSRV
LAST_USED: 2008-01-20 16:12:24.000000
TRYBUY: FALSE
PROC_VENDOR: IBM
PROC_BRAND: POWER5+ QCM
PROC_TYPE: 4
PROC_MODEL:
PROC_COUNT: 2
    ROLE: SERVER
ROLE_OVERRIDE: USERREPORTED
ROLE_EFFECTIVE: SERVER
VALUE_UNITS: 50
VALUE_FROM_TABLE: YES
    PVU: 100
SCAN_ERROR : NO
API_CLIENT: NO
PVU_AGNOSTIC: NO
HYPERVISOR: VMWARE
    GUID: 01.2e.1c.80.e5.04-
        .11.da.aa.ab.00.-
        15.58.0b.d9.47
VERSION: 6
RELEASE: 3
LEVEL: 1
VENDOR_D: IBM(R)
BRAND_D: POWER5(TM) QCM
TYPE_D: Quad-core Module
MODEL_D: All Existing

```

## Field descriptions

### PRODUCT

Rollup of license types into products at the level presented in the **QUERY PVUESTIMATE** command. Possible values are PRODTSMEE, PRODTSM, PRODSAM, PRODMAIL, PRODDDB, PRODSYSB, PRODSpace, PRODSAN, PRODERP, or blank.

### LICENSE\_NAME

The license assigned to this node.

### NODE\_NAME

The node name.

### LAST\_USED

Date and time the identified node last connected to the system under this license.

### TRYBUY

Indicates if running under try and buy mode. Possible values are TRUE or FALSE.

### PROC\_VENDOR

Processor vendor name as reported by the client.

### PROC\_BRAND

Processor brand name as reported by the client.

### PROC\_TYPE

Processor type as reported by the client. This value also reflects the number of cores. Example values are 1=SINGLE CORE, 2=DUO CORE, and 4=QUAD CORE.

### PROC\_MODEL

Processor model as reported by the client.



**PROC\_COUNT**

Processor quantity.

**ROLE** Node role. Possible values are CLIENT, SERVER, or OTHER.

**ROLE\_OVERRIDE**

Override value specified in the **UPDATE NODE** command.

**ROLE\_EFFECTIVE**

Actual role based on the values in the ROLE and ROLE\_OVERRIDE fields.

**VALUE\_UNITS**

Assigned processor value unit (PVU) for the processor.

**PVU** Calculated PVU value.

$PVU \text{ per node} = \text{number of processors per node} * \text{processor type} * \text{pvu value}$

where the processor type represents the number of cores, and the pvu value is the value defined for the processor type in the IBM PVU table.

**VALUE\_FROM\_TABLE**

Flag that indicates whether the PVU was calculated based on the IBM PVU table. Possible values are YES or NO. If NO, a value of 100 is applied for each node defined as a server. If no role is defined for a node, the role of server is assumed for purposes of PVU calculation.

**SCAN\_ERROR**

Flag that indicates whether license information was reported by client. Possible values are YES or NO.

**API\_CLIENT**

Flag that indicates an API application. Possible values are YES or NO.

**PVU\_AGNOSTIC**

Flag indicating that the client version release level is earlier than Tivoli Storage Manager V6.3. If the version is earlier than 6.3, valid PVU metrics are not expected. Possible values are YES or NO.

**HYPERVISOR**

Name of the virtual machine software as reported by the client.

**GUID** Globally Unique Identifier (GUID) of the computer where the node is located. The GUID is obtained from the node table.

**VERSION**

Version of client.

**RELEASE**

Release of client.

**LEVEL**

Level of client.

**VENDOR\_D**

Processor vendor display value from the PVU table.

**BRAND\_D**

Processor brand display value from the PVU table.

**TYPE\_D**

Processor type display value from the PVU table.

**MODEL\_D**

Processor model display value from the PVU table.

## Example: Obtain role and PVU-related information

The following example shows partial results for a selected node, including PVU-related information and role information. Possible roles are CLIENT, SERVER, or OTHER. PVU is calculated only for nodes defined as servers.

```
select * from nodes
```

```
ROLE: CLIENT  
ROLE_0: USEREPORTED  
PVENDOR: INTEL  
PBRAND: INTEL  
PTYPE: 4  
PMODEL:  
PCOUNT: 1  
HYPERVISOR:  
PAPI: NO  
SCANERROR: NO
```

---

## SET commands

Use the **SET** commands to specify values that affect many different Tivoli Storage Manager operations.

- “SET ACCOUNTING (Set accounting records on or off)” on page 1080
- “SET ACTLOGRETENTION (Set the retention period or the size of the activity log)” on page 1081
- “SET ALERTACTIVEDURATION (Set the duration of an active alert)” on page 1083
- “SET ALERTCLOSEDDURATION (Set the duration of a closed alert)” on page 1084
- “SET ALERTEMAIL (Set the alert monitor to email alerts to administrators)” on page 1085
- “SET ALERTEMAILFROMADDR (Set the email address of the sender)” on page 1086
- “SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name)” on page 1087
- “SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port)” on page 1088
- “SET ALERTINACTIVEDURATION (Set the duration of an inactive alert)” on page 1090
- “SET ALERTMONITOR (Set the alert monitor to on or off)” on page 1091
- “SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email )” on page 1089
- “SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts)” on page 1092
- “SET ARCHIVERETENTIONPROTECTION (Activate data retention protection)” on page 1093
- “SET ARREPLRULEDEFAULT (Set the server replication rule for archive data)” on page 1095
- “SET AUTHENTICATION (Set password authentication)” on page 1097
- “SET BKREPLRULEDEFAULT (Set the server replication rule for backup data)” on page 1098
- “SET CLIENTACTDURATION (Set the duration period for the client action)” on page 1101
- “SET CONFIGMANAGER (Specify a configuration manager)” on page 1102
- “SET CONFIGREFRESH (Set managed server configuration refresh)” on page 1103
- “SET CONTEXTMESSAGING (Set message context reporting on or off)” on page 1104
- “SET CPUINFOREFRESH (Refresh interval for the client workstation information scan)” on page 1105
- “SET CROSSDEFINE (Specifies whether to cross-define servers)” on page 1106
- “SET DBRECOVERY (Set the device class for automatic backups)” on page 1107
- “SET DEDUPVERIFICATIONLEVEL (Set the percentage of extents to verify)” on page 1109
- “SET DEFAULTAUTHENTICATION (Set the default authentication method for **REGISTER NODE** and **REGISTER ADMIN** commands)” on page 1111

- “SET DISSIMILARPOLICIES (Enable the policies on the target replication server to manage replicated data)” on page 1113
- “SET DRMACTIVEDATASTGPOOL (Specify the active-data pools to be managed by DRM)” on page 1115
- “SET DRMCHECKLABEL (Specify label checking)” on page 1116
- “SET DRMCMDFILENAME (Specify the name of a file to contain commands)” on page 1117
- “SET DRMCOPYSTGPOOL (Specify the copy storage pools to be managed by DRM)” on page 1118
- “SET DRMCOURIERNAME (Specify the courier name)” on page 1119
- “SET DRMDBBACKUPEXPIREDDAYS (Specify DB backup series expiration)” on page 1120
- “SET DRMFILEPROCESS (Specify file processing)” on page 1122
- “SET DRMINSTRPREFIX (Specify the prefix for recovery instructions file names)” on page 1123
- “SET DRMNOTMOUNTABLENAME (Specify the not mountable location name)” on page 1125
- “SET DRMPLANPREFIX (Specify a prefix for recovery plan file names)” on page 1126
- “SET DRMPLANVPOSTFIX (Specify replacement volume names)” on page 1128
- “SET DRMPRIMSTGPOOL (Specify the primary storage pools to be managed by DRM)” on page 1129
- “SET DRMRPFEXPIREDDAYS (Set criteria for recovery plan file expiration)” on page 1130
- “SET DRMVAULTNAME (Specify the vault name)” on page 1132
- “SET EVENTRETENTION (Set the retention period for event records)” on page 1133
- “SET FAILOVERHLADDRESS (Set a failover high level address)” on page 1134
- “SET INVALIDPWLIMIT (Set the number of invalid logon attempts)” on page 1135
- “SET LDAPPASSWORD (Set the LDAP password for the server)” on page 1137
- “SET LDAPUSER (Specify the user ID for the LDAP directory server)” on page 1138
- “SET LICENSEAUDITPERIOD (Set license audit period)” on page 1140
- “SET MAXCMDRETRIES (Set the maximum number of command retries)” on page 1141
- “SET MAXSCHEDESESSIONS (Set maximum scheduled sessions)” on page 1142
- “SET MINPWLENGTH (Set minimum password length)” on page 1144
- “SET MONITORINGADMIN (Set the name of the monitoring administrator)” on page 1146
- “SET MONITOREDSEVERGROUP (Set the group of monitored servers)” on page 1145
- “SET NODEATRISKINTERVAL (Specifies at-risk mode for an individual node)” on page 1147
- “SET PASSEXP (Set password expiration date)” on page 1149
- “SET PRODUCTOFFERING (Set the product offering that is licensed to your enterprise)” on page 1151
- “SET QUERYSCHEDPERIOD (Set query period for polling client nodes)” on page 1153

- “SET RANDOMIZE (Set randomization of scheduled start times)” on page 1154
- “SET REPLRECOVERDAMAGED (Specify whether damaged files are recovered from a replication server)” on page 1158
- “SET REGISTRATION (Set open or closed registration)” on page 1156
- “SET REPLRETENTION (Set the retention period for replication records)” on page 1161
- “SET REPLSERVER (Set the target replication server)” on page 1163
- “SET RETRYPERIOD (Set time between retry attempts)” on page 1165
- “SET SCHEDMODES (Select a central scheduling mode)” on page 1166
- “SET SERVERHLADDRESS (Set the high-level address of a server)” on page 1169
- “SET SERVERLLADDRESS (Set the low-level address of a server)” on page 1170
- “SET SERVERNAME (Specify the server name)” on page 1171
- “SET SERVERPASSWORD (Set password for server)” on page 1172
- “SET SPREPLRULEDEFAULT (Set the server replication rule for space-managed data)” on page 1173
- “SET SSLKEYRINGPW (Set the SSL key ring password)” on page 1175
- “SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)” on page 1176
- “SET STATUSMONITOR (Specifies whether to enable status monitoring)” on page 1178
- “SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)” on page 1180
- “SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)” on page 1182
- “SET SUBFILE (Set subfile backup for client nodes)” on page 1184
- “SET SUMMARYRETENTION (Set number of days to keep data in activity summary table)” on page 1185
- “SET TAPEALERTMSG (Set tape alert messages on or off)” on page 1186
- “SET TOCLOADRETENTION (Set load retention period for table of contents)” on page 1187
- “SET VMATRISKINTERVAL (Specifies the at-risk mode for an individual VM filesystem)” on page 1188

## 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 dsmacnt.log.

The environment variable, DSMSERV\_ACCOUNTING\_DIR, specifies the directory where the accounting file is located.

### 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 312. 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

At server installation, activity log management is retention-based, and the retention period is set to 30 days.

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.

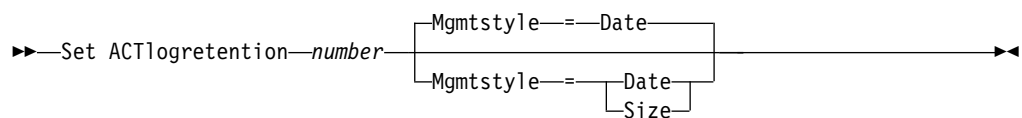
Alternatively, you can choose to limit the total size of the activity log to control the amount of space occupied by the activity log. The server will periodically remove the oldest activity log records until the activity log size no longer exceeds the configured maximum size allowed.

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.

**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 60 days. Issue the command:

```
set actlogretention 60
```

**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 313. 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 ALERTACTIVEDURATION (Set the duration of an active alert)

Use this command to specify how long an alert remains active before it becomes inactive. If an active alert is triggered again, the duration is restarted.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTACTiveduration—*number\_mins*—◄◄

### Parameters

#### *number\_mins* (Required)

Specifies the number of minutes that an alert remains active before it becomes inactive. Specify a value from 1 to 20160. The initial server default value is 480 minutes.

### Set the duration of an active alert to one day

Issue the following command to specify that alerts remain active for 1440 minutes before they change to inactive status:

```
set alertactiveduration 1440
```

### Related commands

Table 314. Commands related to SET ALERTACTIVEDURATION

Command	Description
“QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804	Displays information about monitoring alerts and server status settings.
“SET ALERTINACTIVEDURATION (Set the duration of an inactive alert)” on page 1090	Specifies how long an alert remains inactive before it is closed.
“SET ALERTCLOSEDDURATION (Set the duration of a closed alert)” on page 1084	Specifies how long an alert remains closed before it is deleted.
“SET ALERTMONITOR (Set the alert monitor to on or off)” on page 1091	Specifies whether alert monitoring is set to on or off.
“SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts)” on page 1092	Specifies how often the alert monitor updates and prunes alerts from the database.

## SET ALERTCLOSEDDURATION (Set the duration of a closed alert)

Use this command to specify how long an alert remains closed before it is deleted.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTClosedduration—*number\_mins*—◄◄

### Parameters

#### *number\_mins* (Required)

Specifies the number of minutes that an alert remains closed before it is deleted. Setting the value to 0 causes alerts to be deleted immediately after they are closed. Specify a value from 0 to 99999. The default value is set to 60 minutes when the Tivoli Storage Manager server database is initially formatted.

### Delete alerts two hours after they are closed

Specify that alerts remain closed for 120 minutes before they are deleted:

```
set alertclosedduration 120
```

### Related commands

Table 315. Commands related to SET ALERTCLOSEDDURATION

Command	Description
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"SET ALERTACTIVEDURATION (Set the duration of an active alert)" on page 1083	Specifies how long an alert remains active before it is moved to inactive status.
"SET ALERTINACTIVEDURATION (Set the duration of an inactive alert)" on page 1090	Specifies how long an alert remains inactive before it is closed.
"SET ALERTMONITOR (Set the alert monitor to on or off)" on page 1091	Specifies whether alert monitoring is set to on or off.
"SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts)" on page 1092	Specifies how often the alert monitor updates and prunes alerts from the database.

## SET ALERTEMAIL (Set the alert monitor to email alerts to administrators)

Use this command to enable alerts to be sent to specified administrators by email.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
➤➤—Set ALERTEMail— ON  OFF—➤➤
```

### Parameters

#### ON

Specifies that alerts can be sent to specified administrators by email.

#### OFF

Specifies that alerts cannot be sent to specified administrators by email. When the Tivoli Storage Manager server database is initially formatted, the **ALERTEMAIL** setting is set to OFF.

### Enable alerts to be sent to the administrator when they occur

Enable alerts to be sent by email by issuing the following command:

```
SET ALERTEMAIL ON
```

### Related commands

Table 316. Commands related to SET ALERTEMAIL

Command	Description
“QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804	Displays information about monitoring alerts and server status settings.
“SET ALERTEMAILFROMADDR (Set the email address of the sender)” on page 1086	Specifies the email address of the alert sender.
“SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name)” on page 1087	Specifies the SMTP mail server host name that is used to send alerts by email.
“SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port)” on page 1088	Specifies the SMTP mail server port that is used to send alerts by email.
“SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email )” on page 1089	Specifies the administrators that want to receive alert summaries by email.

## SET ALERTEMAILFROMADDR (Set the email address of the sender)

Use this command to specify the email address of the alert sender.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTEMAILFRomaddr—*email\_address*—◀◀

### Parameters

#### *email\_address* (Required)

Specifies the email address of the sender. Email addresses are in the form of *name@domain*. Email names, including the address, cannot exceed 64 characters in length, and the domain name cannot exceed 255 characters in length.

### Specify the email address of the alert sender

Specify the email address of the sender by issuing the following command:

```
set alertemailfromaddr djadmin@mydomain.com
```

### Related commands

Table 317. Commands related to SET ALERTEMAILFROMADDR

Command	Description
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"SET ALERTEMAIL (Set the alert monitor to email alerts to administrators)" on page 1085	Enables alerts to be sent by email to specified administrators.
"SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name)" on page 1087	Specifies the SMTP mail server host name that is used to send alerts by email.
"SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port)" on page 1088	Specifies the SMTP mail server port that is used to send alerts by email.
"SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email )" on page 1089	Specifies the administrators that want to receive alert summaries by email.

## SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name)

Use this command to specify the Simple Mail Transfer Protocol (SMTP) mail server host name that is used to send the alert email.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTEMAILSMTPHost—*host\_name*—◄◄

### Parameters

#### *host\_name* (Required)

Specifies the SMTP mail server host name.

### Specify the host name for the SMTP mail server as mail.domain.com

Specify mail.domain.com as the SMTP mail server, by issuing the following command:

```
set alertemailsmtp host mail.domain.com
```

### Related commands

Table 318. Commands related to SET ALERTEMAILSMTPHOST

Command	Description
“SET ALERTEMAIL (Set the alert monitor to email alerts to administrators)” on page 1085	Enables alerts to be sent by email to specified administrators.
“SET ALERTEMAILFROMADDR (Set the email address of the sender)” on page 1086	Specifies the email address of the alert sender.
“SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port)” on page 1088	Specifies the SMTP mail server port that is used to send alerts by email.
“SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email )” on page 1089	Specifies the administrators that want to receive alert summaries by email.

## SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port)

Use this command to specify the port number for the SMTP mail server. This mail server is used to send the alerts by email.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTEMAILSMTPPort—*tcp\_port*—————►►

### Parameters

#### *tcp\_port* (Required)

Specifies the port number of the SMTP mail server. Specify a value of 1 through 32767. The default port number is 25.

### Specify the port number of the SMTP mail server

Specify port number 450 as your SMTP mail server by issuing the following command:

```
set alertemailsmtpport 450
```

### Related commands

Table 319. Commands related to SET ALERTEMAILSMTPPORT

Command	Description
“SET ALERTEMAIL (Set the alert monitor to email alerts to administrators)” on page 1085	Enables alerts to be sent by email to specified administrators.
“SET ALERTEMAILFROMADDR (Set the email address of the sender)” on page 1086	Specifies the email address of the alert sender.
“SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name)” on page 1087	Specifies the SMTP mail server host name that is used to send alerts by email.
“SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email )” on page 1089	Specifies the administrators that want to receive alert summaries by email.

## SET ALERTSUMMARYTOADMINS (Set the list of administrators to receive alert summaries by email )

Use this command to specify the administrators that want to receive alert summaries by email, every hour.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTSUMMARYToadmins admin\_name ◀◀

### Parameters

#### *admin\_name* (Required)

Specifies the administrator name that wants to receive alert summaries by email. You can specify up to three administrator names by separating them with commas and no intervening spaces.

### Specify two administrators to receive alert summaries

Specify that administrators HARRY and COLIN want to receive alert summaries, by issuing the following command:

```
set alertsummarytoadmins HARRY,COLIN
```

### Related commands

Table 320. Commands related to SET ALERTSUMMARYTOADMINS

Command	Description
“SET ALERTEMAIL (Set the alert monitor to email alerts to administrators)” on page 1085	Enables alerts to be sent by email to specified administrators.
“SET ALERTEMAILFROMADDR (Set the email address of the sender)” on page 1086	Specifies the email address of the alert sender.
“SET ALERTEMAILSMTPHOST (Set the SMTP mail server host name)” on page 1087	Specifies the SMTP mail server host name that is used to send alerts by email.
“SET ALERTEMAILSMTPPORT (Set the SMTP mail server host port)” on page 1088	Specifies the SMTP mail server port that is used to send alerts by email.

## SET ALERTINACTIVEDURATION (Set the duration of an inactive alert)

Use this command to specify how long an alert remains inactive. After the inactive duration is past, the alert is closed.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTINactiveduration—*number\_mins*—————►►

### Parameters

#### *number\_mins* (Required)

Specifies the number of minutes that an alert remains inactive before it is closed. You can specify a value in the range 1 - 20160. The initial server default value is 480 minutes.

### Change alert status from inactive to closed after 60 minutes

Issue the following command to specify that an alert remains in inactive status for 60 minutes before it changes to closed status:

```
set alertinactiveduration 60
```

### Related commands

Table 321. Commands related to SET ALERTINACTIVEDURATION

Command	Description
"SET ALERTACTIVEDURATION (Set the duration of an active alert)" on page 1083	Specifies how long an alert remains active before it is moved to inactive status.
"SET ALERTCLOSEDDURATION (Set the duration of a closed alert)" on page 1084	Specifies how long an alert remains closed before it is deleted.
"SET ALERTMONITOR (Set the alert monitor to on or off)" on page 1091	Specifies whether alert monitoring is set to on or off.
"SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts)" on page 1092	Specifies how often the alert monitor updates and prunes alerts from the database.



## SET ALERTMONITOR (Set the alert monitor to on or off)

Use this command to turn the alert monitor on or off.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► Set ALERTMONITOR  OFF  ON ◀◀

### Parameters

#### ON

Specifies that the Tivoli Storage Manager server monitors alerts.

#### OFF

Specifies that the Tivoli Storage Manager server does not monitor alerts. When the Tivoli Storage Manager server database is initially formatted, the alert monitoring setting is set to OFF.

### Turn on alert monitoring

Turn on alert monitoring by issuing the following command:

```
set alertmonitor on
```

### Related commands

Table 322. Commands related to SET ALERTMONITOR

Command	Description
"SET ALERTACTIVEDURATION (Set the duration of an active alert)" on page 1083	Specifies how long an alert remains inactive before it is closed.
"SET ALERTINACTIVEDURATION (Set the duration of an inactive alert)" on page 1090	Specifies how long an alert remains inactive before it is closed.
"SET ALERTCLOSEDDURATION (Set the duration of a closed alert)" on page 1084	Specifies how long an alert remains closed before it is deleted.
"SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts)" on page 1092	Specifies how often the alert monitor updates and prunes alerts from the database.

## SET ALERTUPDATEINTERVAL (Set how often the alert monitor updates and prunes alerts)

Use this command to specify how often the alert monitor updates and prunes alerts that are stored in the Tivoli Storage Manager server database.

During this check interval, the alert monitor examines each alert on the server and completes the following actions:

- The alert monitor determines whether the active or inactive durations elapsed. If the specified duration elapses, the alert status is updated to the next state. For example:
  - Active to Inactive
  - Inactive to Closed
- If an alert is closed for the duration that is specified by the **SET ALERTCLOSEDDURATION** command, the alert is deleted.

You can use the **QUERY MONITORSETTINGS** command to determine whether alert monitoring is on. Use the **SET ALERTMONITOR** command to turn on alert monitoring.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set ALERTUPDateinterval—*number\_mins*—————►►

### Parameters

#### *number\_mins* (Required)

Specifies the length of time, in minutes, that the monitor waits before alerts are updated and pruned on the server. Specify a value from 1 to 9999. The server has an initial default value of 10 minutes.

### Set alert update interval to 60 minutes

Specify that alerts are updated every hour by issuing the following command:

```
set alertupdateinterval 60
```

### Related commands

Table 323. Commands related to SET ALERTUPDATEINTERVAL

Command	Description
“SET ALERTACTIVEDURATION (Set the duration of an active alert)” on page 1083	Specifies how long an alert remains active before it is moved to inactive status.
“SET ALERTINACTIVEDURATION (Set the duration of an inactive alert)” on page 1090	Specifies how long an alert remains inactive before it is closed.
“SET ALERTCLOSEDDURATION (Set the duration of a closed alert)” on page 1084	Specifies how long an alert remains closed before it is deleted.
“SET ALERTMONITOR (Set the alert monitor to on or off)” on page 1091	Specifies whether alert monitoring is set to on or off.

## 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 is 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 324. 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.

Table 324. Commands related to SET ARCHIVERETENTIONPROTECTION (continued)

Command	Description
DELETE FILESPACE	Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group.
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 ARREPLRULEDEFAULT (Set the server replication rule for archive data)

Use this command to set the server replication rule for archive data.

**Restriction:** The replication rule that you set with this command is applied only if file space rules and client node rules for archive data are set to DEFAULT.

Issue this command on the server that acts as a source for replicated data.

You can specify a normal-priority replication rule or a high-priority replication rule. In a replication process that includes both normal-priority and high-priority data, high-priority data is replicated first. Before you specify a rule, consider the order in which you want the data to be replicated.

For example, suppose that your client nodes contain archive data and backup data. Replication of the archive data is a higher priority than the backup data. To prioritize the archive data, issue the **SET ARREPLRULEDEFAULT** command and specify the ALL\_DATA\_HIGH\_PRIORITY replication rule. To prioritize the backup data, issue the **SET BKREPLRULEDEFAULT** command and specify the ALL\_DATA replication rule for backup data. The ALL\_DATA rule for backup data replicates backup data with a normal priority.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►► Set ARREPLRuledefault { ALL_DATA | ALL_DATA_HIGH_PRIORITY | NONE } ►►
```

### Parameters

#### ALL\_DATA

Replicates archive data with a normal priority.

#### ALL\_DATA\_HIGH\_PRIORITY

Replicates archive data with a high priority.

#### NONE

Archive data is not replicated.

### Example: Set the server replication rule for archive data

Set up the default rule for archive data to replicate with a high priority.

```
set arreplruledefault all_data_high_priority
```

### Related commands

Table 325. Commands related to SET ARREPLRULEDEFAULT

Command	Description
QUERY FILESPACE	Displays information about data in file spaces that belong to a client.

Table 325. Commands related to SET ARREPLRULEDEFAULT (continued)

Command	Description
QUERY NODE	Displays partial or complete information about one or more clients.
QUERY REPLICATION	Displays information about node replication processes.
QUERY REPLRULE	Displays information about node replication rules.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
SET BKREPLRULEDEFAULT	Specifies the server node-replication rule for backup data.
SET SPREPLRULEDEFAULT	Specifies the server node-replication rule for space-managed data.
UPDATE FILESPACE	Changes file-space node-replication rules.
UPDATE REPLRULE	Enables or disables replication rules.
VALIDATE REPLICATION	Verifies replication for file spaces and data types.

## 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 that authenticate to the IBM Tivoli Storage Manager server do not require a password to access the server. Administrators and client nodes that authenticate to an LDAP directory server are not affected.

### 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 326. Command related to SET AUTHENTICATION

Command	Description
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.

## SET BKREPLRULEDEFAULT (Set the server replication rule for backup data)

Use this command to set the server replication rule for backup data.

**Restriction:** The replication rule that you set with this command is applied only if file space rules and client node rules for backup data are set to DEFAULT.

Issue this command on the server that acts as a source for replicated data.

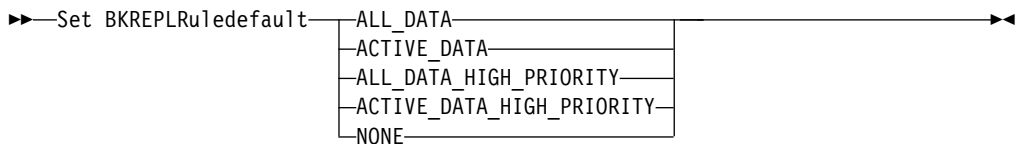
You can specify normal-priority replication rules or high-priority replication rules. In a replication process that includes both normal and high-priority data, high-priority data is replicated first. Before you specify a rule, consider the order in which you want the data to be replicated.

For example, suppose that your client nodes contain archive data and active backup data. Replication of the active backup data is a higher priority than the archive data. To prioritize the backup data, issue the **SET BKREPLRULEDEFAULT** command and specify the **ACTIVE\_DATA\_HIGH\_PRIORITY** replication rule. To prioritize the archive data, issue the **SET ARREPLRULEDEFAULT** command and specify the **ALL\_DATA** replication rule for archive data. The **ALL\_DATA** rule for archive data replicates archive data with a normal priority.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### ALL\_DATA

Replicates active and inactive backup data. The data is replicated with normal priority.

#### ACTIVE\_DATA

Replicates active backup data. The data is replicated with normal priority.



**Attention:** If you specify `ACTIVE_DATA` and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the `FORCERECONCILE=YES` parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

#### **ALL\_DATA\_HIGH\_PRIORITY**

Replicates active and inactive backup data. Data is replicated with a high priority.

#### **ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the `ACTIVE_DATA` replication rule except data is replicated with a high priority.

#### **NONE**

Backup data is not replicated.

### **Example: Set the server replication rule for backup data**

Set up the default rule for backup data to replicate only active data and to replicate the data with a high priority.

```
set bkreplruledefault active_data_high_priority
```

### **Related commands**

*Table 327. Commands related to SET BKREPLRULEDEFAULT*

<b>Command</b>	<b>Description</b>
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 REPLICATION	Displays information about node replication processes.
QUERY REPLRULE	Displays information about node replication rules.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
SET ARREPLRULEDEFAULT	Specifies the server node-replication rule for archive data.
SET REPLRETENTION	Specifies the retention period for replication history records.

Table 327. Commands related to SET BKREPLRULEDEFAULT (continued)

<b>Command</b>	<b>Description</b>
SET SPREPLRULEDEFAULT	Specifies the server node-replication rule for space-managed data.
UPDATE FILESPACE	Changes file-space node-replication rules.
UPDATE REPLRULE	Enables or disables replication rules.
VALIDATE REPLICATION	Verifies replication for file spaces and data types.

## 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 328. 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

►► Set CONFIGManager  Off  ON ►►

### 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 329. 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 330. 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

```
►► Set CONTEXTmessaging { ON | Off } ►►
```

### 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 331. Commands related to SET CONTEXTMESSAGING*

Command	Description
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.

## SET CPUINFOREFRESH (Refresh interval for the client workstation information scan)

Use this command to specify the number of days between client scans of workstation information that is used to estimate the processor value unit (PVU).

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set CPUINFOREFRESH—*days*—————►►

### Parameters

*days* (Required)

Specifies the number of days between scans for client devices. To retrieve the current setting, issue the **QUERY STATUS** command. The possible values are 1 - 9999. The default is 180.

### Example: Set the amount of time before the next refresh to 90 days

```
SET CPUINFOREFRESH 90
```

### Related commands

Table 332. Commands related to SET CPUINFOREFRESH

Command	Description
QUERY PVUESTIMATE	Displays an estimate of the client-devices and server-devices being managed.

## 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 333. 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)

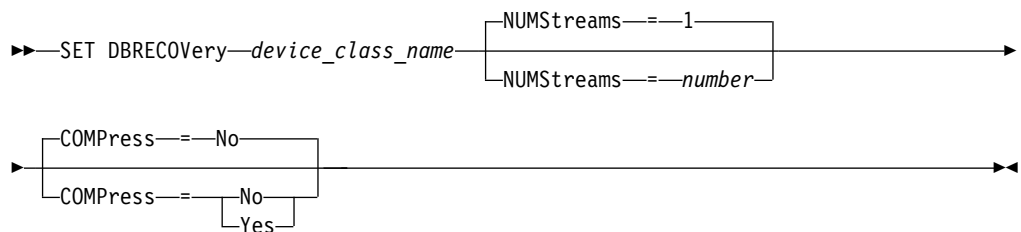
Use this command to specify the device class and number of data streams to be used for automatic backups.

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



### Parameters

#### *device\_class\_name* (Required)

Specifies the device class to use for database backups.

#### **NUMStreams**

Specifies the number of parallel data movement streams to use when you back up the database. The default value is 1, and the maximum number is 4. Increasing this value causes a corresponding increase in the number of database backup sessions to be used and in the number of drives to be used for the device class. A **NUMSTREAMS** value that is specified in the **BACKUP DB** command overrides any value set in the **SET DBRECOVERY** command. The **NUMSTREAMS** value is used for all types of database backups.

If a value is specified that is greater than the number of drives available for the device class, the number of available drives are used. The available drives are those defined to the device class by the **MOUNTLIMIT** parameter or by the number of online drives for the specified device class. The session is displayed in the **QUERY SESSION** output.

If you increase the number of streams, more volumes are used from the corresponding device class for this operation. Using more volumes might improve the speed of the database backups, but at the cost of more volumes that are not fully used.

#### **COMPRESS**

Specifies whether volumes are compressed during database backup processing. This parameter is optional. The default value is No. You can specify one of the following values:

**No** Specifies that the volumes created by the **BACKUP DB** command are not compressed.

**Yes**

Specifies that the volumes created by the **BACKUP DB** command are compressed.

If you specify the **COMPRESS** parameter on the **BACKUP DB** command, it overrides any value that is set in the **SET DBRECOVERY** command. Otherwise, the value that is set in the **SET DBRECOVERY** command is used.

**Restrictions:**

- Use caution when you specify the **COMPRESS** parameter. Using compression during database backups can reduce the size of the backup files. However, compression can increase the time that is required to complete database backup processing.
- Do not back up compressed data to tape. If your system environment stores database backups on tape, set the **COMPRESS** parameter to No in the **SET DBRECOVERY** and **BACKUP DB** commands.

**Example: Specify a device class for database backups**

Specify the DBBACK device class for database backups. Issue the command:

```
set dbrecovery dback
```

**Example: Specify a device class and number of streams for database backups**

Specify the DBBACK device class for database backups, and specify that the backup is to use two data movement streams. Issue the command:

```
set dbrecovery dback numstreams=2
```

**Related commands**

*Table 334. Commands related to SET DBRECOVERY*

Command	Description
BACKUP DB	Backs up the 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 DEDUPVERIFICATIONLEVEL (Set the percentage of extents to verify)

Use this command to verify extents sent to the server during client-side data deduplication.

A rogue application that resides on a client system and that imitates the client, API, or GUI application can initiate an attack on the server. To reduce server vulnerability to such attacks, you can specify a percentage of client extents for the server to verify.

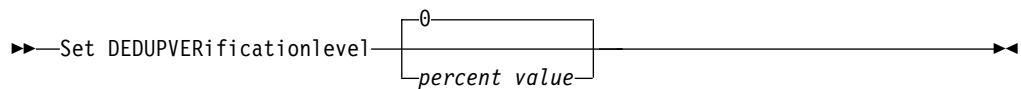
If the server detects that a security attack is in progress, the current session is canceled. In addition, the setting of the **DEDUPLICATION** parameter on the **REGISTER NODE** command is changed. The setting is changed from **CLIENTORSERVER** to **SERVERONLY**. The **SERVERONLY** setting disables client-side data deduplication for that node.

The server also issues a message that a potential security attack was detected and that client-side data deduplication was disabled for the node. If client-side data deduplication is disabled, all other client operations (for example, backup operations) continue. Only client-side data deduplication is disabled. If client-side data deduplication is disabled for a node because a potential attack was detected, the server deduplicates the data that is eligible for client-side data deduplication.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *percent\_value* (Required)

Specify an integer value 0 - 100 to indicate the percentage of client extents to be verified. A value of 0 indicates that no client extents are verified. The default for this command is 0.

#### Tips:

- Verifying extents consumes processing power and adversely affects server performance. For optimal performance, do not specify values greater than 10 for this command.
- To display the current value for **SET DEDUPVERIFICATIONLEVEL**, issue the **QUERY STATUS** command.

### Example: Specify a minimum level of data deduplication verification

To specify that 1% of extents created during client-side data deduplication are verified, issue the following command:

```
set dedupverificationlevel 1
```

## Example: Turn off data deduplication verification

To specify that none of the extents created during client-side data deduplication are verified, issue the following command:

```
set dedupverificationlevel 0
```

## Related commands

*Table 335. Commands related to SET DEDUPVERIFICATIONLEVEL*

Command	Description
DEFINE STGPOOL	Defines a storage pool as a named collection of server storage media.
QUERY CONTENT	Displays information about files in a storage pool volume.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REGISTER NODE	Defines a client node to the server and sets options for that user.
UPDATE NODE	Changes the attributes that are associated with a client node.
UPDATE STGPOOL	Changes the attributes of a storage pool.

## SET DEFAULTAUTHENTICATION (Set the default authentication method for REGISTER NODE and REGISTER ADMIN commands)

Use this command to set the default password authentication method for nodes and administrators that are the result of **REGISTER NODE** or **REGISTER ADMIN** commands.

If you specify LDAP, you establish the default value for authenticating to an external directory for any new **REGISTER NODE** or **REGISTER ADMIN** commands. This command makes it easier to register nodes or administrators when you use an LDAP directory server.

**Tip:** The default authentication setting can be overwritten when the authentication method is specified in a **REGISTER NODE** or **REGISTER ADMIN** command.

### Privilege class

To issue this command you must have system privilege.

### Syntax

```
➤—SET DEFAULTAUTHentication—┬─Local──┬──────────────────────────────────────────────────────────────────────────────────▶  
                               └─Ldap──┘
```

### Parameters

#### Local

Specifies that any future **REGISTER NODE** or **REGISTER ADMIN** commands that you issue use LOCAL as the default authentication parameter value. Locally-authenticated passwords are those stored on the IBM Tivoli Storage Manager server. The passwords authenticated locally are not case sensitive.

#### Ldap

Specifies that any future **REGISTER NODE** or **REGISTER ADMIN** commands that you issue use LDAP as the default authentication parameter value. LDAP-authenticated passwords are those stored on an LDAP directory server and are case sensitive.

### Example: Set the default password authentication value to LDAP

Specify that any **REGISTER NODE** or **REGISTER ADMIN** commands that you issue authenticate passwords with an LDAP directory server.

```
set defaultauthentication ldap
```

### Related commands

Table 336. Commands related to SET DEFAULTAUTHENTICATION

Command	Description
SET LDAPPASSWORD	Sets the password for the LDAPUSER.
SET LDAPUSER	Sets the user who oversees the passwords and administrators on the LDAP directory server.

*Table 336. Commands related to SET DEFAULTAUTHENTICATION (continued)*

<b>Command</b>	<b>Description</b>
SET LDAPUSER	Sets the user who oversees the passwords and administrators on the LDAP directory server.
REGISTER ADMIN	Defines a new administrator without granting administrative authority.
REGISTER NODE	Defines a client node to the server and sets options for that user.

## SET DISSIMILARPOLICIES (Enable the policies on the target replication server to manage replicated data)

Use the **SET DISSIMILARPOLICIES** command to enable the policies that are defined on the target replication server to manage replicated client-node data. If you do not use the policies on the target replication server, replicated client-node data is managed by policies on the source replication server.

Ensure that Tivoli Storage Manager, Version 7.1.1 or later, is installed on the source and target replication servers before you issue this command. Issue this command on the source replication server.

Before you use the policies that are defined on a target replication server, you must issue the **VALIDATE REPLPOLICY** command for that target replication server. This command displays the differences between the policies for the client nodes on the source replication server and policies on the target replication server. You can modify the policies on the target replication server before you enable these policies to manage replicated client-node data.

To obtain the name of the target replication server for which you want to manage data and to check whether the policies on the target replication server are set to ON, use the **QUERY REPLSERVER** command. At installation, the value is set to OFF.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set DISSIMILARPolicies—*target\_server\_name*—

Off
Off
ON

►►

### Parameters

***target\_server\_name* (Required)**

Specifies the name of the target replication server for which you want to enable the policies.

**ON** Specifies that replicated client-node data is managed by the policies that are defined on the target replication server.

**Off**

Specifies that replicated client-node data is managed by the policies that are defined on the source replication server. Off is the default value.

### Example: Use the policies on a target replication server

To managed replicated client-node data from the target replication server, CVTCVS\_LXS\_SRV2, issue the following command on the source replication server:

```
set dissimilarpolicies CVTCVS_LXS_SRV2 on
```

## Related commands

*Table 337. Commands related to SET DISSIMILARPOLICIES*

<b>Command</b>	<b>Description</b>
QUERY REPLSERVER	Displays information about replicating servers.
VALIDATE REPLPOLICY	Verifies the policies on the target replication server.



## 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 **ACTIVEDATASTGPOOL** 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 **ACTIVEDATASTGPOOL** 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 338. Commands related to **SET DRMACTIVEDATASTGPOOL**

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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 DRMPRIMSTGPOOL	Specifies that primary storage pools are managed by DRM.

## SET DRMCKECLABEL (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 DRMCKECLABEL is set to YES.

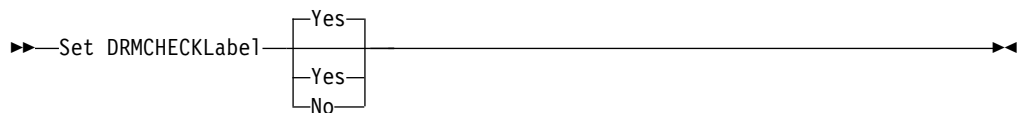
Use the **QUERY DRMSTATUS** command to check the current setting.

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 339. Commands related to SET DRMCKECLABEL

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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 340. Commands related to SET DRMCMDFILENAME

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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 341. Commands related to **SET DRMCOPYSTGPOOL**

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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 342. Commands related to SET DRMCOURIERNAME

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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 exceeds the expiration value set with the **SET DRMDBBACKUPEXPIREDAYS** command and the value that is specified for the **DELgraceperiod** parameter in the **DEFINE SERVER** command. The **DELgraceperiod** parameter applies only to remote database backups. The default value for the **DELgraceperiod** parameter is 5 days. For example, if you set the value for the **SET DRMDBBACKUPEXPIREDAYS** command to 7 days and set the value for the **DELgraceperiod** parameter to 6 days, the remote database backup series does not expire until 13 days elapse.
- 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 that are managed by disaster recovery manager. Specify an integer value 0 - 9999.

### Example: Set the database backup series expiration

Set the database backup series expiration value to 60.

```
set drmdbbackupexpiredays 60
```

## Related commands

*Table 343. Commands related to SET DRMDBBACKUPEXPIREDAYS*

<b>Command</b>	<b>Description</b>
DSMSERV RESTORE DB	Restores a Tivoli Storage Manager database.
MOVE DRMEDIA	Moves DRM media onsite and offsite.
QUERY DRMEDIA	Displays information about disaster recovery volumes.
QUERY DRMSTATUS	Displays DRM system parameters.
QUERY VOLHISTORY	Displays sequential volume history information that has been collected by the server.
DEFINE SERVER	Defines a server for server-to-server communications.

## 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 344. Commands related to SET DRMFILEPROCESS

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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.

**Restriction:** The recovery instructions files must be created on a file system that supports long file names.

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 /drmp1an/primesrv.  
set drminstrprefix /drmp1an/primesrv/

## Related commands

*Table 345. 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 room 123/31.

```
set drmnountable "room 123/31"
```

### Related commands

Table 346. Commands related to SET DRMNOTMOUNTABLENAME

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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.

**Restriction:** The recovery plan file must be created on a file system that supports long file names.

### 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:

/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:

/drmpplan/primsrv

Issue the command:

```
set drmpplanprefix /drmpplan/primsrv/
```

### **Related commands**

*Table 347. Commands related to SET DRMPPLANPREFIX*

<b>Command</b>	<b>Description</b>
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.

### 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 348. 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)

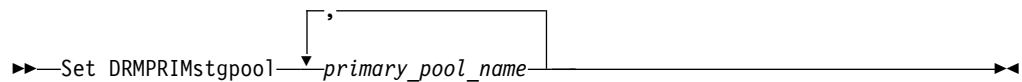
Use this command to specify the names of primary storage pools that you want to recover. If the **PREPARE** command does not include the PRIMSTGPOOL parameter, DRM processes the names specified in this command.

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



### 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 349. 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 that are 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 exceeds the expiration value that is specified with the **SET DRMRPFEXPIREDAYS** command and the value that is specified for the **DELgraceperiod** parameter in the **DEFINE SERVER** command. The default value for the **DELgraceperiod** parameter is 5 days. For example, if you set the value for the **SET DRMRPFEXPIREDAYS** command to 80 days and set the value for the **DELgraceperiod** parameter to 6 days, the recovery plan file does not expire until 86 days elapse.
- The latest recovery plan file is not associated with the most recent database backup series.

For more information about expiration processing, see the **EXPIRE INVENTORY** command.

### 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 0 - 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 350. Commands related to SET DRMRPFEXPIREDAYS*

Command	Description
PREPARE	Creates a recovery plan file.



*Table 350. Commands related to SET DRMRPFEXPIREDAYS (continued)*

<b>Command</b>	<b>Description</b>
QUERY DRMSTATUS	Displays DRM system parameters.
QUERY RPFCONTENT	Displays the contents of 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.
SET DRMDBBACKUPEXPIREDAYS	Specifies criteria for database backup series expiration.
DEFINE SERVER	Defines a server for server-to-server communications.

## 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 DRMVAultname—*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 351. Commands related to SET DRMVAULTNAME

Command	Description
MOVE DRMEDIA	Moves DRM media onsite and offsite.
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 352. Commands related to SET EVENTRETENTION

Command	Description
DELETE EVENT	Deletes event records before 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 FAILOVERHLADDRESS (Set a failover high level address)

Use this command to specify the IP address that a client uses to connect to this server as the secondary replication server during failover, if the address is different from the IP address that is specified for the replication process.

You must specify the address of the server that is used if the high-level address (HLA) is different. This command is required only if you use separate dedicated networks for server-to-server communication and client access.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—SET FAILOVERHLaddress—*high\_level\_address*—◄◄

### Parameters

*high\_level\_address* **(Required)**

Specifies a server HLA as a numeric dotted decimal name or a host name to use during failover. If you specify a host name, a server that can resolve the name to the dotted decimal format must be available.

To remove the failover IP address, issue the command without specifying a value.

### Example: Set a failover high-level address

The name of the HLA that you want to set for failover operations on this server.  
set failoverhladdress server1

### Example: Remove a high-level address

To remove a high-level address for a failover server, issue the following command:  
set failoverhladdress

### Related commands

Table 353. Commands related to QUERY REPLSERVER

Command	Description
"QUERY REPLSERVER (Query a replication server)" on page 881	Displays information about replicating servers.
"REMOVE REPLSERVER (Remove a replication server)" on page 1014	Removes a server from replication.

## SET INVALIDPWLIMIT (Set the number of invalid logon attempts)

Use this command to set the number of invalid logon attempts that are allowed before a node is locked.

The **SET INVALIDPWLIMIT** command also applies to LDAP directory servers that store complex node passwords. LDAP directory servers can limit the number of invalid password attempts independent of the IBM Tivoli Storage Manager server. You might not want to set up the LDAP directory server for invalid attempts for the Tivoli Storage Manager namespace if you use the **SET INVALIDPWLIMIT** command.

### 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 one time, the node is locked by the server. The default is 0.

**Important:** If your password is authenticated with an LDAP directory server, it can be managed by the LDAP server and the Tivoli Storage Manager server. Not all Tivoli Storage Manager server commands affect passwords that authenticate with an LDAP server. For example, the **SET PASSEXP** and **RESET PASSEXP** commands do not affect passwords that authenticate with an LDAP directory server. You can manage your password features through the Tivoli Storage Manager server. If you issued the **SET INVALIDPWLIMIT** command, all Tivoli Storage Manager passwords are controlled by the limit that you set. If you configure the LDAP directory server to limit the number of invalid password attempts, a conflict might occur.

### Example: Define the number of allowed invalid login attempts

Set the number of invalid logon attempts allowed.

```
set invalidpwlimit 6
```

### Related commands

Table 354. Commands related to SET INVALIDPWLIMIT

Command	Description
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.
QUERY NODE	Displays partial or complete information about one or more clients.

*Table 354. Commands related to SET INVALIDPWLIMIT (continued)*

<b>Command</b>	<b>Description</b>
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 LDAPPASSWORD (Set the LDAP password for the server)

Use this command to define a password for the user name or ID that you specify in the **SET LDAPUSER** command.

This user can then gain access to the Tivoli Storage Manager space on the LDAP directory server to issue administrative operations, such as resetting passwords.

**Requirement:** You must define the **LDAPURL** option and issue the **SET LDAPUSER** command before submitting the **SET LDAPPASSWORD** command. If the **LDAPURL** option is not defined when you set the **LDAPPASSWORD**, you must restart the Tivoli Storage Manager server after defining the **LDAPURL** option.

### Privilege class

To issue this command you must have system privilege.

### Syntax

►►—Set LDAPPASSWORD—*ldap\_user\_password*—◄◄

### Parameters

*ldap\_user\_password*

Specifies the password that the server uses when it authenticates to the LDAP directory server. The maximum length of the password is 64 characters. If you have equal signs within your password, you must contain the whole password within quotation marks in the **SET LDAPPASSWORD** command. You can use the following characters:

a b c d e f g h i j k l m n o p q r s t u v w x y z  
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
0 1 2 3 4 5 6 7 8 9  
~ ! @ # \$ % ^ & \* \_ - + = ` | ( ) { } [ ] : ; < > , . ? /

### Example: Set the LDAP password

```
set ldappassword LdAp20&12PaSsWoRd
```

### Example: Set an LDAP password that includes an equal sign

```
set ldappassword "LdAp=LastWoRd"
```

### Related commands

Table 355. Commands related to SET LDAPPASSWORD

Command	Description
AUDIT LDAPDIRECTORY	Audit an IBM Tivoli Storage Manager-controlled namespace on an LDAP directory server.
SET DEFAULTAUTHENTICATION	Specifies the default password authentication method for any REGISTER NODE or REGISTER ADMIN commands.
SET LDAPUSER	Sets the user who oversees the passwords and administrators on the LDAP directory server.

## SET LDAPUSER (Specify the user ID for the LDAP directory server)

Use this command to specify the ID of a user or account that can conduct Tivoli Storage Manager administrative operations when it accesses the LDAP directory server. The user ID must have access to the base distinguished name (Base DN) on the LDAP directory server.

This command determines which ID issues administrative operations, such as resetting node passwords or entering user accounts, on the LDAP directory server. You must grant Base DN access to the **LDAPUSER**. The ID that is specified in the **SET LDAPUSER** command must be able to create, delete, and modify objects under the Base DN.

To change the **LDAPUSER**, reissue the **SET LDAPUSER** command.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set LDAPUser—*ldap\_user\_dn*—————►►

### Parameters

*ldap\_user\_dn*

Specifies the user DN that is used when the Tivoli Storage Manager server needs to issue administrative commands to the Tivoli Storage Manager name space on the LDAP directory server. If the user DN contains spaces or equal signs (=), enclose the entire DN in quotation marks. The maximum length of the DN is 256 characters.

**Important:** Enclose the user DN in single or double quotation marks.

If you do not know the DN when you enter the **SET LDAPUSER** command, contact your LDAP administrator. If you use Windows Active Directory, you can determine the DN by using one of the following methods:

#### Base your query on the name attribute

Issue the **dsquery user -name tsmuser** command.

#### Base your query on the user principal name (UPN)

Issue the **dsquery user -upn tsmuser@tsmdev.storage.us.ibm.com** command.

#### Base your query on the SAM account name

Issue the **dsquery user -samid tsmuser** command.

### Example: Specify the LDAPUSER

Issue the following **SET LDAPUSER** command to work with an LDAP Directory Server:

```
set ldapuser "uid=jackspratt,ou=users,o=ibm.com,c=us"
```



This command sets the user DN with a user ID (uid) of jackspratt, with organizational unit (ou) as users, organization (o) as ibm.com, and country (c) as the United States.

Issue the following **SET LDAPUSER** command to work with a Windows Active Directory server and other directory servers:

```
set ldapuser "cn=Jack Spratt,cn=users,dc=us,dc=ibm,dc=com"
```

This command sets the user DN with a common name (cn) of Jack Spratt and users, and a domain component (dc) of us, ibm, and com.

## Related commands

*Table 356. Commands related to SET LDAPUSER*

Command	Description
AUDIT LDAPDIRECTORY	Audit an IBM Tivoli Storage Manager-controlled namespace on an LDAP directory server.
SET DEFAULTAUTHENTICATION	Specifies the default password authentication method for any REGISTER NODE or REGISTER ADMIN commands.
SET LDAPPASSWORD	Sets the password for the LDAPUSER.

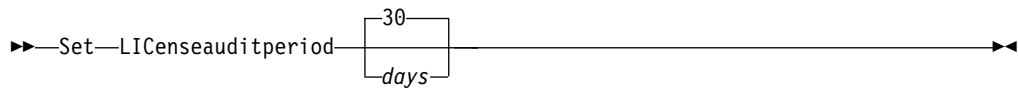
## SET LICENSEAUDITPERIOD (Set license audit period)

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



### 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 357. Commands related to SET LICENSEAUDITPERIOD

Command	Description
AUDIT LICENSES	Verifies 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 license with the 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 358. 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 MAXSCHEDESESSIONS (Set maximum scheduled sessions)

Use this command to set the number of sessions that the server can use to process 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 some 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, however, 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.

**Tip:** If the maximum number of scheduled sessions do not coincide with the percentage that you set in the **SET MAXSCHEDESESSIONS** command, run the **SET MAXSCHEDESESSIONS** command again. Look in the **MAXSESSIONS** option and determine the number that is specified there. If the **MAXSESSIONS** option number changed and you did not issue the **SET MAXSCHEDESESSIONS** command since the change, the maximum number of scheduled sessions can change.

### Set a maximum of 20 sessions for scheduled activities

The **MAXSESSIONS** option has a value of 80. If you want no more than 20 sessions to be available for scheduled activity, set the percentage to 25.

```
set maxschedsessions 25
```

### Related commands

Table 359. Commands related to SET MAXSCHEDESESSIONS

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. The default value for minimum password length is set to 0.

### Example: Set the minimum password length

Set the minimum password length to 5 characters.

```
set minpwlenght 5
```

### Related commands

Table 360. 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 MONITOREDSEVERGROUP (Set the group of monitored servers)

Use this command to set the group of servers that are being monitored for alerts and status. You can also use this command to change or remove the group of monitored servers.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—Set MONITOREDSEVERGroup group_name—►►
```

### Parameters

*group\_name*

Specifies the Tivoli Storage Manager server group name that contains all monitored servers. You can remove a monitored server group name by issuing the command without specifying a value, or by specifying an empty value (""). Any existing monitoring for alerts and status from remote servers is ended.

### Set the name of a monitored server group

Set the name of a monitored server group SUBS, by issuing the following command:

```
set monitoredservergroup subs
```

### Remove the name of a monitored server group

Remove the monitored server group, by issuing the following command:

```
set monitoredservergroup
```

### Related commands

Table 361. Commands related to SET MONITOREDSEVERGROUP

Command	Description
"DEFINE SERVERGROUP (Define a server group)" on page 322	Defines a new server group.
"DEFINE GRPMEMBER (Add a server to a server group)" on page 231	Defines a server as a member of a server group.
"DELETE GRPMEMBER (Delete a server from a server group)" on page 416	Deletes a server from a server group.
"QUERY SERVERGROUP (Query a server group)" on page 910	Displays information about server groups.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"SET MONITORINGADMIN (Set the name of the monitoring administrator)" on page 1146	Set the name of the monitoring administrator.

## SET MONITORINGADMIN (Set the name of the monitoring administrator)

Use this command to set the name of the monitoring administrator that is used to connect to the servers in the monitored server group.

To display the name of the monitored server group, issue the **QUERY MONITORSETTINGS** command.

The administrator name that you specify must match the name of an existing administrator, otherwise the command fails.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►► Set MONITORINGADMIN admin\_name ►►

### Parameters

*admin\_name*

Specifies administrator names. You can remove names by issuing the command without specifying a value, or by specifying an empty value ("").

### Set the monitoring administrator name

Set the name of the monitoring administrator to MONADMIN, by issuing the following command:

```
set monitoringadmin monadmin
```

### Remove the monitoring administrator name

Remove the monitoring administrator, by issuing the following command:

```
set monitoringadmin ""
```

### Related commands

Table 362. Commands related to SET MONITORINGADMIN

Command	Description
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"SET MONITOREDSEVERGROUP (Set the group of monitored servers)" on page 1145	Set the group of monitored servers.



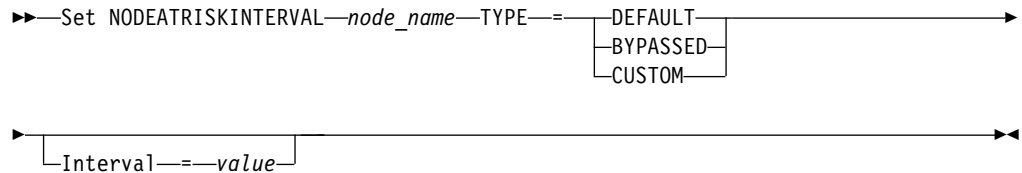
## SET NODEATRISKINTERVAL (Specifies at-risk mode for an individual node)

Use this command to adjust the at-risk evaluation mode for an individual node.

### 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 client node that you want to update.

#### TYPE (Required)

Specifies the at-risk evaluation type. Specify one of the following values:

##### DEFAULT

Specifies that the node is evaluated with the same interval that was specified for the nodes classification by the **SET STATUSATRISKINTERVAL** command. The value is either system or applications, or VM, and is determined by the status monitor.

For example, you can specify TYPE = DEFAULT, which allows the status monitor to go ahead and classify the node automatically. Then the interval that is used, is the interval that was defined for that classification by the **SET STATUSATRISKINTERVAL** command.

##### BYPASSED

Specifies that the node is not evaluated for at-risk status by the status monitor. The at risk status is also reported as bypassed to the Operations Center.

##### CUSTOM

Specifies that the node is evaluated with the specified interval, rather than the interval that was specified by the **SET STATUSATRISKINTERVAL** command.

#### Interval

Specifies the amount of time, in hours, between client backup activity before the status monitor considers the client to be at risk. You can specify an integer in the range 6 - 8808. You must specify this parameter when TYPE = CUSTOM. You do not specify this parameter when TYPE = BYPASSED or TYPE = DEFAULT. The interval value for all client types is set to 24 at server installation.

### Set node name to use a custom 90 day at-risk interval

Set the at-risk interval for a node named *fred* to 90 days.

```
set nodeatriskinterval fred type=custom interval=2160
```

## Bypass the at-risk interval evaluation

Bypass the at-risk interval checking for a node named *bob*.

```
set nodeatriskinterval bob type=bypassed
```

### Related commands

Table 363. Commands related to `set nodeatriskinterval`

Command	Description
“SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)” on page 1176	Specifies whether to enable client at-risk activity interval evaluation
“SET VMATRISKINTERVAL (Specifies the at-risk mode for an individual VM filesystem)” on page 1188	Sets the at-risk mode for a VM filesystem
“QUERY MONITORSTATUS (Query the monitoring status)” on page 807	Displays information about monitoring alerts and server status settings.
“QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804	Displays information about monitoring alerts and server status settings.
“SET STATUSMONITOR (Specifies whether to enable status monitoring)” on page 1178	Specifies whether to enable status monitoring.
“SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)” on page 1180	Specifies the refresh interval for status monitoring.
“SET STATUSSKIPFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)” on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
“QUERY NODE (Query nodes)” on page 817	Displays partial or complete information about one or more clients.
“QUERY FILESPACE (Query one or more file spaces)” on page 772	Displays information about data in file spaces that belong to a client.

## SET PASSEXP (Set password expiration date)

Use this command to set the expiration period for administrator and client node passwords. You can either set a common password expiration period for all administrators and client node passwords or selectively set password expiration periods.

**Restriction:** The **SET PASSEXP** command does not apply to passwords that authenticate with an LDAP directory server.

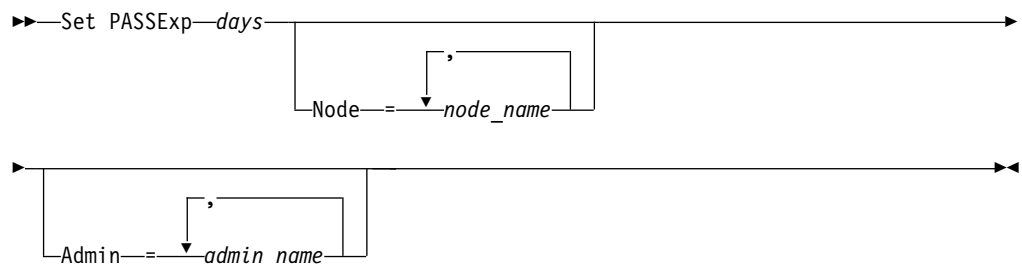
You can override the **SET PASSEXP** setting for one or more nodes by using the **REGISTER NODE** or **UPDATE NODE** command with the **PASSEXP** parameter.

The **NODE** or **ADMIN** parameters must be specified to change the password expiration period for client nodes or administrators with selectively set password expiration periods. 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.

### 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 for which you are setting the password expiration period. 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 364. 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 node 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 that are associated with a client node.

## SET PRODUCTOFFERING (Set the product offering that is licensed to your enterprise)

Use the **SET PRODUCTOFFERING** command to define the IBM Tivoli Storage Manager product offering that is licensed to your enterprise.

The definition is used to determine whether automatic storage capacity measurement calculations are required and made available for use by the IBM License Metric Tool (ILMT). Run this command only if you are using ILMT to determine license consumption.

For product offerings where automatic storage capacity measurement calculations are made available for use by ILMT, the parameter also defines which capacity measurement approach is used for those calculations.

The same storage capacity information is made available to ILMT on a weekly interval. After an applicable product offering is defined by using this command, Tivoli Storage Manager makes the current capacity calculation for that offering available to the ILMT. After the initial capacity calculation is made available to ILMT, Tivoli Storage Manager updates the value weekly.

### Privilege class

To run this command, you must have system privilege.

### Syntax

►►—SET PRODUCTOFFERING—*product\_offering*—►►

### Parameters

#### *product\_offering* (Required)

Specifies a product offering. The maximum length of the text string is 255 characters. The following options are available:

**SSAM** Specifies that the product offering licensed in your enterprise is System Storage Archive Manager. Capacity measurements for this product offering are not calculated automatically or made available for use by ILMT.

**TSM** Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager. This product offering uses a processor value unit (PVU) licensing metric. Capacity measurements for this product offering are not applicable.

#### **TSMEE**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Extended Edition. This product offering uses a PVU licensing metric. Capacity measurements for this product offering are not applicable.

#### **TSMSUR**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Suite for Unified Recovery. Capacity measurements for this product offering are calculated automatically and made available for use by ILMT.

**TSMSURENTRY**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Suite for Unified Recovery Entry. Capacity measurements for this product offering are calculated automatically and made available for use by ILMT.

**TSMSURARChive**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Suite for Unified Recovery - Archive. Capacity measurements for this product offering are calculated automatically and made available for use by ILMT.

**TSMSURPROtectier**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Suite for Unified Recovery - ProtecTier. Capacity measurements for this product offering are calculated automatically and made available for use by ILMT.

**TSMSURFROntend**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Suite for Unified Recovery - FrontEnd. Capacity measurements for this product offering are calculated automatically and made available for use by ILMT.

**TSMSURENTRYFROntend**

Specifies that the product offering licensed in your enterprise is Tivoli Storage Manager Suite for Unified Recovery Entry - FrontEnd. Capacity measurements for this product offering are calculated automatically and made available for use by ILMT.

**CLEAR**

No product offering is specified.

### **Example: Set the product offering to Tivoli Storage Manager (TSM)**

```
set productoffering TSM
```

### **Related commands**

*Table 365. Commands related to SET PRODUCTOFFERING*

<b>Command</b>	<b>Description</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.

## 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 366. 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. When the scheduling mode is not set to polling, 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 367. Commands related to SET RANDOMIZE*

<b>Command</b>	<b>Description</b>
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 authentication method is based on the **DEFAULTAUTHENTICATION** server setting.
- SSL (Secure Sockets Layer) is required for registering client nodes that authenticate with an LDAP directory server.
- 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

```
►► Set REGISTRATION { Closed | Open } ►►
```

### 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.

### **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 368. Command related to SET REGISTRATION*

<b>Command</b>	<b>Description</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.

## SET REPLRECOVERDAMAGED (Specify whether damaged files are recovered from a replication server)

Use this command to enable the system-wide recovery of damaged files from a target replication server. If this setting is turned on, the node replication process can be configured to detect damaged files on the source replication server and replace them with undamaged files from the target replication server.

The **REPLRECOVERDAMAGED** system parameter affects all file recovery processes across all replication processes for all nodes and file spaces. File recovery is possible only if Tivoli Storage Manager, Version 7.1.1 or later, is installed on the source and target replication servers, and if the node data was replicated before the file damage occurred.

To display the current setting, use the **QUERY STATUS** command.

When you install the server, the default setting is ON.

If you upgrade the server and no damaged files are detected, the default setting is ON.

If you upgrade the server and damaged files are detected, the parameter is set to OFF, and a message is issued to indicate that the recovery of damaged files is disabled. The OFF setting prevents the server from scanning database tables for damaged objects that can be recovered. Prevention of the scan is necessary in case many damaged files are detected. In that case, a scan can take a considerable amount of time, and should be scheduled when use of server resources is at a minimum. When you are ready to start the scan and recover damaged files, you must issue the **SET REPLRECOVERDAMAGED** command and specify the ON setting. After the server successfully completes the scan, the **REPLRECOVERDAMAGED** system parameter is set to ON.

The following table describes how the **REPLRECOVERDAMAGED** system parameter and other parameters affect the recovery of damaged, replicated files.

Table 369. Settings that affect the recovery of damaged files.

Setting for the <b>REPLRECOVERDAMAGED</b> system parameter	Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REPLICATE NODE</b> command	Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REGISTER NODE</b> and <b>UPDATE NODE</b> commands	Result
OFF	YES, NO, or not specified	YES or NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.
OFF	ONLY	YES or NO	An error message is displayed because files cannot be recovered when the <b>REPLRECOVERDAMAGED</b> system parameter is set to OFF.

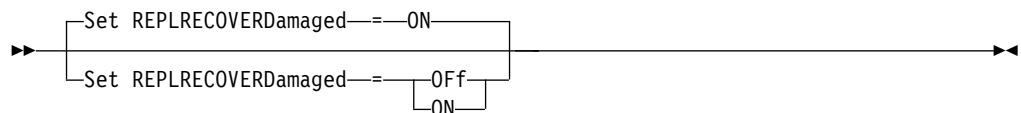
Table 369. Settings that affect the recovery of damaged files (continued).

Setting for the REPLRECOVERDAMAGED system parameter	Value of the RECOVERDAMAGED parameter on the REPLICATE NODE command	Value of the RECOVERDAMAGED parameter on the REGISTER NODE and UPDATE NODE commands	Result
ON	YES	YES or NO	During node replication, standard replication occurs and damaged files are recovered from the target replication server.
ON	NO	YES or NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.
ON	ONLY	YES or NO	Damaged files are recovered from the target replication server, but standard node replication does not occur.
ON	Not specified	YES	During node replication, standard replication occurs and damaged files are recovered from the target replication server.
ON	Not specified	NO	During node replication, standard replication occurs and damaged files are not recovered from the target replication server.

## Privilege class

To issue this command, you must have system privilege.

## Syntax



## Parameters

**ON** Specifies that node replication is enabled to recover damaged files from a target replication server.

**OFF**

Specifies that node replication is not enabled to recover damaged files from a target replication server.

## Example: Enable recovery of damaged files

To specify a system-wide setting that enables the server to recover damaged files from a target replication server, issue the following command:

```
set replrecoverdamaged on
```

## Related commands

*Table 370. Commands related to SET REPLRECOVERDAMAGED*

Command	Description
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REGISTER NODE	Defines a client node to the server and sets options for that user.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
UPDATE NODE	Changes the attributes that are associated with a client node.

## SET REPLRETENTION (Set the retention period for replication records)

To maintain adequate information about replication processes, you can use this command to adjust the length of time that the source replication server retains replication records in its database. The **SET REPLRETENTION** command specifies the retention period for client-node replication records in the source replication-server database. You can use client node replication records to monitor running and completed processes.

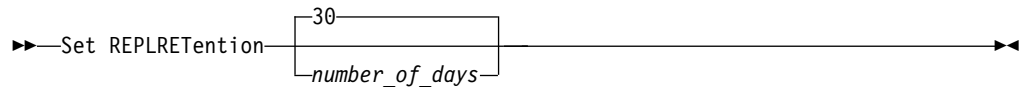
A replication record is created when **REPLICATE NODE** command processing is started. By default, Tivoli Storage Manager retains client-node replication records for 30 calendar days. A calendar day consists of 24 hours, from midnight to midnight. For example, suppose that the retention period is two calendar days. If a replication process completes at 11:00 p.m. on day *n*, a record of that process is retained for 25 hours until midnight on day *n+1*. To display the retention period for replication records, issue the **QUERY STATUS** command on the source replication server.

Issue the **SET REPLRETENTION** command on the server that acts as a source for replicated data.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

*number\_of\_days* (Required)

The number of days that the source replication server retains replication records. You can specify an integer 0 - 9999. The default value is 30.

### Example: Set a retention period for client-node replication records

You want to retain client-node replication records for 10 days.

```
set replretention 10
```

### Related commands

Table 371. Commands related to SET REPLRETENTION

Command	Description
QUERY REPLICATION	Displays information about node replication processes.
QUERY REPLNODE	Displays information about the replication status of a client node.
QUERY REPLRULE	Displays information about node replication rules.

*Table 371. Commands related to SET REPLETENTION (continued)*

<b>Command</b>	<b>Description</b>
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.



## SET REPLSERVER (Set the target replication server)

Use this command to set the name of a target replication server. You can also use this command to change or remove a target replication server.

Issue this command on the server that acts as a source for replicated data.

To display the name of a target replication server, issue the **QUERY STATUS** command on a source replication server.

### Important:

- The server name that you specify with this command must match the name of an existing server definition. It must also be the name of the server to be used as the target replication server. If the server name specified by this command does not match the server name of an existing server definition, the command fails.
- Use care when you are changing or removing a target replication server. If you change a target replication server, replicated client-node data is sent to a different target replication server. If you remove a target replication server, client node data is not replicated.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set REPLSERVER—*target\_server\_name*—►►

### Parameters

#### *target\_server\_name* (Required)

Specifies the name of the target replication server. The name that you specify must match the name of an existing server. The maximum length of a name is 64 characters.

To remove a target replication server, issue the command without specifying a value.

**Note:** If you do not want to continue replicating data, you can remove the node replication configuration after you remove the target replication server.

### Example: Set a target replication server

The name of the server that you want to set as the target replication server is SERVER1.

```
set replserver server1
```

### Related commands

Table 372. Commands related to SET REPLSERVER

Command	Description
DEFINE SERVER	Defines a server for server-to-server communications.
QUERY SERVER	Displays information about servers.

*Table 372. Commands related to SET REPLSERVER (continued)*

<b>Command</b>	<b>Description</b>
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
UPDATE SERVER	Updates information about a server.
REMOVE REPLNODE	Removes a node from replication.
REMOVE REPLSERVER	Removes a server from replication.

## 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 373. 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.

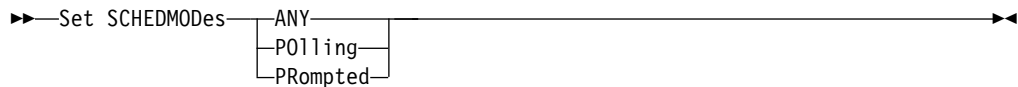
Use this command with the **SET RETRYPERIOD** command to regulate the time and the number of retry attempts to process a failed command.

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 374. Command related to **SET SCHEDMODES**

Command	Description
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.

Table 374. Command related to SET SCHEDMODES (continued)

Command	Description
SET RETRYPERIOD	Specifies the time between retry attempts by the client scheduler.

## SET SCRATCHPADRETENTION (Set scratch pad retention time)

Use this command to set the amount of time for which scratch pad entries are retained.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—SET SCRATCHPADRETENTION—*days*—◄◄

### Parameters

#### *days* (Required)

Specifies the number of days that a scratchpad entry is retained after the last update to the scratchpad entry. You can enter an integer in the range 1 - 9999.

### Example: Retain scratch pad entries for 367 days after they are updated

```
set scratchpadretention 367
```

### Related commands

Table 375. Commands related to SET SCRATCHPADRETENTION

Command	Description
DEFINE SCRATCHPADENTRY	Creates a line of data in the scratch pad.
DELETE SCRATCHPADENTRY	Deletes a line of data from the scratch pad.
QUERY SCRATCHPADENTRY	Displays information that is contained in the scratch pad.
UPDATE SCRATCHPADENTRY	Updates data on a line in the scratch pad.

## 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**. You must use the **SET SERVERHLADDRESS** command for all automatic client deployments.

### 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 376. 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 377. 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. 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 378. 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. Setting the password protects the master encryption key. Data on the server is only encrypted if you set the password. After you set the password, you cannot remove the password.

### 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 password

Set the password for HQ\_SERVER to agave.

```
set serverpassword agave
```

### Related commands

Table 379. 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 SPREPLRULEDEFAULT (Set the server replication rule for space-managed data)

Use this command to set the server replication rule for space-managed data.

**Restriction:** The replication rule that you set with this command is applied only if file space rules and client node rules for space-managed data are set to DEFAULT.

Issue this command on the server that acts as a source for replicated data.

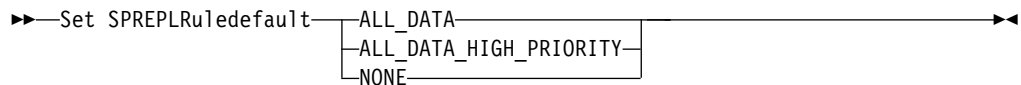
You can specify a normal-priority replication rule or a high-priority replication rule. In a replication process that includes both normal and high-priority data, high-priority data is replicated first. Before you specify a rule, consider the order in which you want the data to be replicated.

For example, suppose that your client nodes contain space-managed data and backup data. Replication of the space-managed data is a higher priority than the backup data. To prioritize the space-managed data, issue the **SET SPREPLRULEDEFAULT** command and specify the ALL\_DATA\_HIGH\_PRIORITY replication rule. To prioritize the backup data, issue the **SET BKREPLRULEDEFAULT** command and specify the ALL\_DATA replication rule for backup data. The ALL\_DATA rule for backup data replicates backup data with a normal priority.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### ALL\_DATA

Replicates space-managed data with a normal priority.

#### ALL\_DATA\_HIGH\_PRIORITY

Replicates space-managed data with a high priority.

#### NONE

Space-managed data is not replicated.

### Example: Set the server replication rule for space-managed data

Set up the default rule for space-managed data to replicate with a high priority.

```
set spreplruledefault all_data_high_priority
```

### Related commands

Table 380. Commands related to SET BKREPLRULEDEFAULT

Command	Description
QUERY FILESPACE	Displays information about data in file spaces that belong to a client.

Table 380. Commands related to SET BKREPLRULEDEFAULT (continued)

Command	Description
QUERY NODE	Displays partial or complete information about one or more clients.
QUERY REPLICATION	Displays information about node replication processes.
QUERY REPLRULE	Displays information about node replication rules.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
REPLICATE NODE	Replicates data in file spaces that belong to a client node.
SET ARREPLRULEDEFAULT	Specifies the server node-replication rule for archive data.
SET BKREPLRULEDEFAULT	Specifies the server node-replication rule for backup data.
UPDATE FILESPACE	Changes file-space node-replication rules.
UPDATE REPLRULE	Enables or disables replication rules.
VALIDATE REPLICATION	Verifies replication for file spaces and data types.

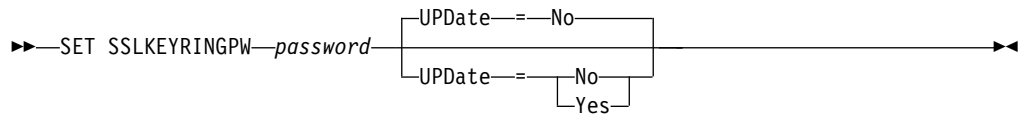
## SET SSLKEYRINGPW (Set the SSL key ring password)

Use this command to provide the key database file password to the server. You can also use it to update the key database file password.

### Privilege class

You must have system privileges to issue this command.

### Syntax



### Parameters

#### *password* (Required)

Specifies the password to use to access the key database file (`cert.kdb`). This parameter is required and is limited to 64 characters.

#### UPDate

Specifies whether to allow the key database file password to be updated. This parameter is optional. The default is **NO**. Possible values are:

**No** When specified, notifies Tivoli Storage Manager that the key database file password was changed outside of the server. The server validates the new password before recording it for use on a subsequent startup.

#### Yes

Specifies to use the password that you defined in the **password** parameter, along with the stored password to change the key database file password. The password is changed immediately in the key database file and Tivoli Storage Manager uses it on a subsequent startup.

### Example: Query all background processes

Update the existing stored password with the new password. Restart the server to use the new password.

```
set sslkeyring newpassword update=yes
```

### Related commands

Table 381. Commands related to SET SSLKEYRINGPW

Command	Description
DELETE KEYRING	Deletes password information in the certificate key database.
QUERY SSLKEYRINGPW	Displays the Secure Sockets Layer (SSL) key database file password.

## SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)

Use this command to adjust the backup activity interval that is used when the status monitor assesses whether clients are at risk.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
➤➤ Set STATUSATRISKINTERVAL TYPE = [ALl | APplications | VM | SYstems] Interval = value ➤➤
```

### Parameters

#### TYPE (Required)

Specifies the type of client that should be evaluated. Specify one of the following values:

##### ALl

Specify this setting for all client types.

##### APplications

Specify this setting for only application client types.

##### VM

Specify this setting for virtual system clients types.

##### SYstems

Specify this setting for systems client types.

#### Interval (Required)

Specifies the amount of time, in hours, between client activity before the status monitor considers the client to be at risk. You can specify an integer in the range 6 - 8808. The interval value for all client types is set to 24 at server installation.

### Set systems to use a two-week at-risk interval

Set the at-risk interval check for systems client types to 2 weeks.

```
set statusriskinterval type=systems interval=336
```

### Related commands

Table 382. Commands related to

Command	Description
"DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)" on page 326	Defines a status monitoring threshold.
"DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442	Deletes a status monitoring threshold.
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.

*Table 382. Commands related to (continued)*

<b>Command</b>	<b>Description</b>
“QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804	Displays information about monitoring alerts and server status settings.
“QUERY STATUSTHRESHOLD (Query status monitoring thresholds)” on page 932	Displays information about a status monitoring thresholds.
“SET STATUSMONITOR (Specifies whether to enable status monitoring)” on page 1178	Specifies whether to enable status monitoring.
“SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)” on page 1180	Specifies the refresh interval for status monitoring.
“SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)” on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
“UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)” on page 1389	Changes the attributes of an existing status monitoring threshold.

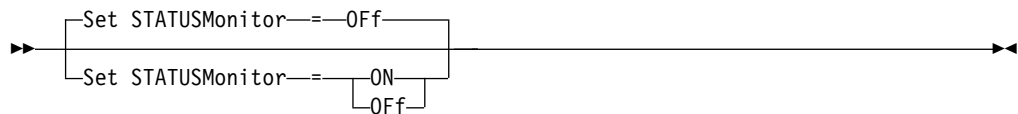
## SET STATUSMONITOR (Specifies whether to enable status monitoring)

Use this command to enable and disable status monitoring. Turning status monitoring on for the first time also sets the default threshold values, and increases the event record retention to at least 14 days.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

**ON** Specifies that the status monitoring is turned on. The first time that you set status monitoring to ON, it sets all the default threshold values that are specified in the **DEFINE STATUSTHRESHOLD** and **UPDATE STATUSTHRESHOLD** commands. It also sets the retention value for event records to at least 14 days. For example, when you turn status monitoring on, the default values for primary storage pool utilization is automatically set to display a warning when the threshold value reaches 80%, and an error when the threshold reaches 90% utilization.

#### OFF

Specifies that the status monitoring is turned off. Off is the default value.

### Enable status monitoring

Set status monitoring to on to enable status monitoring.

```
set statusmonitor on
```

### Related commands

Table 383. Commands related to SET STATUSMONITOR

Command	Description
"DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)" on page 326	Defines a status monitoring threshold.
"DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442	Deletes a status monitoring threshold.
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932	Displays information about a status monitoring thresholds.



*Table 383. Commands related to SET STATUSMONITOR (continued)*

<b>Command</b>	<b>Description</b>
“SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)” on page 1176	Specifies whether to enable client at-risk activity interval evaluation
“SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)” on page 1180	Specifies the refresh interval for status monitoring.
“SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)” on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
“UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)” on page 1389	Changes the attributes of an existing status monitoring threshold.

## SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)

Use this command to specify the number of minutes between status monitoring server queries.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

►►—Set STATUSREFreshinterval—*minutes*—►►

### Parameters

#### *minutes* (Required)

Specifies the approximate number of minutes between status monitoring server queries. You can specify an integer in the range 1 - 2440. The default value is 5.

#### Restrictions:

- In a storage environment that is monitored by the Operations Center, set the same refresh interval on the hub and spoke servers. If you use different intervals, the Operations Center can show inaccurate information for spoke servers.
- Short status refresh intervals use more space in the server database and might require more processor and disk resources. For example, decreasing the interval by half doubles the required database and archive log space. Long intervals reduce the currency of Operations Center data but better suit a high-latency network configuration.
- A status refresh interval of less than 5 minutes can cause the following issues:
  - Operations Center data that is supposed to be refreshed after the defined interval takes a longer time to be refreshed.
  - Operations Center data that is supposed to be refreshed almost immediately when a related change occurs in the storage environment also takes a longer time to be refreshed.

### Set the refresh interval for status monitoring

Specify that the server status is queried every 6 minutes, by issuing the following command:

```
set statusrefreshinterval 6
```

### Related commands

Table 384. Commands related to SET STATUSREFRESHINTERVAL

Command	Description
"DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)" on page 326	Defines a status monitoring threshold.
"DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442	Deletes a status monitoring threshold.

Table 384. Commands related to SET STATUSREFRESHINTERVAL (continued)

Command	Description
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932	Displays information about a status monitoring thresholds.
"SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176	Specifies whether to enable client at-risk activity interval evaluation
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)" on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
"UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389	Changes the attributes of an existing status monitoring threshold.

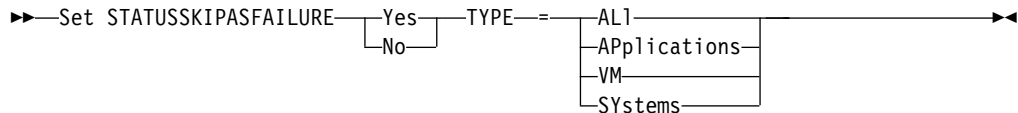
## SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)

Use this command to enable the status monitor to consider clients as at risk when evaluating the status for each client.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### State (Required)

Specifies whether to enable the check for skipped files during the last backup. This check signifies that the client is at-risk if any files were skipped. Client data that is skipped or not backed up properly is considered at risk.

#### Yes

Specifies that the server evaluates whether a client is at risk.

**No** Specifies that the server does not evaluate whether a client is at risk.

#### TYPE (Required)

Specifies the type of client that should be evaluated. Specify one of the following values:

#### ALL

Specify this setting for all client types.

#### Applications

Specify this setting for only application client types.

#### VM

Specify this setting for virtual system clients types.

#### Systems

Specify this setting for systems client types.

### Disable at-risk evaluation for virtual system client types

Disable the at-risk evaluation for virtual systems client types by issuing the following command:

```
set statusskipasfailure off type=vm
```

### Related commands

Table 385. Commands related to SET STATUSSKIPASFAILURE

Command	Description
"DEFINE STATUSTHRESHOLD (Define a status monitoring threshold)" on page 326	Defines a status monitoring threshold.
"DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442	Deletes a status monitoring threshold.

*Table 385. Commands related to SET STATUSSKIPASFAILURE (continued)*

<b>Command</b>	<b>Description</b>
"QUERY MONITORSTATUS (Query the monitoring status)" on page 807	Displays information about monitoring alerts and server status settings.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932	Displays information about a status monitoring thresholds.
"SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176	Specifies whether to enable client at-risk activity interval evaluation
"SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178	Specifies whether to enable status monitoring.
"SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)" on page 1180	Specifies the refresh interval for status monitoring.
"UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389	Changes the attributes of an existing status monitoring threshold.

## 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 workstation, the SUBFILECACHEPATH and SUBFILECACHESIZE options must be specified in the client's options file (dsm.opt). If you are using a Windows client, you must also specify the SUBFILEBACKUP option.

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 386. 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 387. 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 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, the server can retrieve diagnostic information from a tape or library device and display it using ANR messages. When disabled, the server 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 server.

**OFF**

Specifies that diagnostic information will not be reported to the server.

### Example: Set tape alert messages on

Allow the server to receive diagnostic information messages.

```
set tapealertmsg on
```

### Related commands

Table 388. 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 389. Commands related to SET TOCLOADRETENTION

Command	Description
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.

## SET VMATRISKINTERVAL (Specifies the at-risk mode for an individual VM filespace)

Use this command to adjust the at-risk evaluation mode for an individual VM filespace.

### 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

```
▶▶ Set VMATRISKINTERVAL—node_name—fsid—TYPE—=—  
|—DEFAULT—|  
|—BYPASSED—|  
|—CUSTOM—|  
▶▶  
|—Interval—=—value—|  
▶▶
```

### Parameters

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

Specifies the name of the client node, that owns the VM filespace, that you want to update.

#### *fsid* (Required)

Specifies the filespace ID of the client node that you want to update.

#### TYPE (Required)

Specifies which at-risk evaluation mode the status monitor should use when evaluating the at-risk classification for the specified nodes VM filespace. Specify one of the following values:

##### DEFAULT

Specifies that the VM filespace is evaluated with the same interval that was specified for the **SET STATUSATRISKINTERVAL** command.

##### BYPASSED

Specifies that the VM filespace is not evaluated for at-risk status by the status monitor. The at-risk status is also reported as bypassed to the Operations Center.

##### CUSTOM

Specifies that the VM filespace is evaluated with the specified interval, rather than the interval that was specified for the **SET STATUSATRISKINTERVAL** command.

#### Interval

Specifies the amount of time, in hours, between client backup activity before the status monitor considers the client to be at risk. You can specify an integer in the range 6 - 8808. You must specify this parameter when TYPE = CUSTOM. You do not specify this parameter when TYPE = BYPASSED or TYPE = DEFAULT. The interval value for all client types is set to 24 at server installation.

## Set node name to use a custom 90 day at-risk interval

Set the at-risk interval for a node named *charlievm* (filesystem ID 50) on datacenter node named *alice* to use a 90 day at-risk interval. You can issue the **QUERY FILESPACE** command to determine the filesystem ID for the VM.

```
set vmatriskinterval alice 50 type=custom interval=2160
```

## Bypass the at-risk interval evaluation

Exclude the VM called *davevm* (filesystem ID 213) on datacenter node named *erin* from at-risk interval checking. You can issue the **QUERY FILESPACE** command to determine the filesystem ID for the VM called *davevm*. Then set the at-risk interval check for the VM as bypassed.

```
set vmatriskinterval erin 213 type=bypassed
```

## Related commands

Table 390. Commands related to `set vmatriskinterval`

Command	Description
“SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)” on page 1176	Specifies whether to enable client at-risk activity interval evaluation
“SET NODEATRISKINTERVAL (Specifies at-risk mode for an individual node)” on page 1147	Sets the at-risk mode and interval for a node
“QUERY MONITORSTATUS (Query the monitoring status)” on page 807	Displays information about monitoring alerts and server status settings.
“QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)” on page 804	Displays information about monitoring alerts and server status settings.
“SET STATUSMONITOR (Specifies whether to enable status monitoring)” on page 1178	Specifies whether to enable status monitoring.
“SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)” on page 1180	Specifies the refresh interval for status monitoring.
“SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)” on page 1182	Specifies whether to use client at-risk skipped files as failure evaluation
“QUERY NODE (Query nodes)” on page 817	Displays partial or complete information about one or more clients.
“QUERY FILESPACE (Query one or more file spaces)” on page 772	Displays information about data in file spaces that belong to a client.

---

## SETOPT (Set a server option for dynamic update)

You can use the **SETOPT** command to update most server options dynamically without stopping and restarting the server. For the **DBDIAGLOGSIZE** option, you must stop and start 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:

- ADMINCOMMTIMEOUT
- ADMINIDLETIMEOUT
- ALLOWREORGINDEX
- ALLOWREORGTABLE
- ARCHLOGCOMPRESS
- CHECKTAPEPOS
- CLIENTDEDUPTXNLIMIT
- COMMTIMEOUT
- DBDIAGLOGSIZE
- DBDIAGPATHFSTHRESHOLD
- DEDUPTIER2FILESIZE
- DEDUPTIER3FILESIZE
- DEDUPREQUIRESBACKUP
- DNSLOOKUP
- EXPINTERVAL
- EXPQUIET
- FSUSEDTHRESHOLD
- IDLETIMEOUT
- LDAPCACHEDURATION
- MAXSESSIONS
- MOVEBATCHSIZE
- MOVESIZETHRESH
- NDMPPREFDATAINTERFACE
- NUMOPENVOLSAALLOWED
- RECLAIMDELAY
- RECLAIMPERIOD
- REORGBEGINTIME
- REORGDURATION
- RESOURCE\_TIMEOUT
- RESTOREINTERVAL
- RETENTIONEXTENSION
- SERVERDEDUPTXNLIMIT
- SHREDDING
- THROUGHPUTDATATHRESHOLD

THROUGHPUTTimethreshold  
TXNGroupmax

*option\_value* **(Required)**

Specifies the value for the server option.

### **Example: Set the maximum number of client sessions**

Update the server option for the maximum number of client sessions to a value of 40.

```
setopt maxsessions 40
```

### **Related commands**

*Table 391. Commands related to SETOPT*

<b>Command</b>	<b>Description</b>
QUERY OPTION	Displays information about server options.
QUERY SYSTEM	Displays details about the 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.

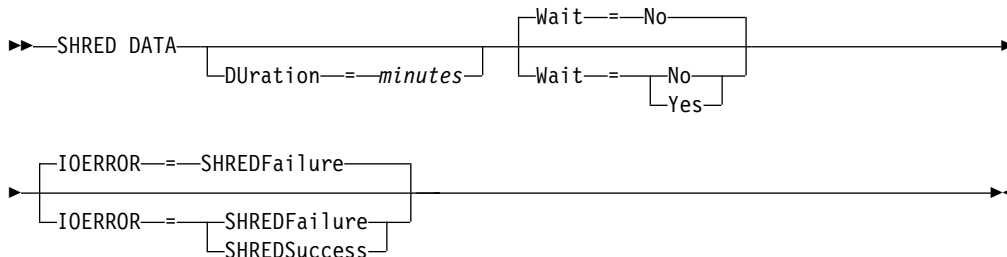
This command creates a background process that can be cancelled with the **CANCEL PROCESS** command. To display information on background processes, use the **QUERY PROCESS** command.

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 392. Commands related to SHRED DATA*

<b>Command</b>	<b>Description</b>
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

```
▶▶ SUSPend EXPOrt [ * ] export_identifier ▶▶
```

### 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 393. Commands related to **SUSPEND EXPORT**

Command	Description
CANCEL EXPORT	Deletes a suspended export operation.
EXPORT NODE	Copies client node information to external media or directly to another server.
EXPORT SERVER	Copies all or part of the server to external media or directly to another server.
QUERY EXPORT	Displays the export operations that are currently running or suspended.
RESTART EXPORT	Restarts a suspended export operation.



---

## UNLOCK commands

Use the **UNLOCK** commands to reestablish access after an object was locked.

- “UNLOCK ADMIN (Unlock an administrator)” on page 1196
- “UNLOCK NODE (Unlock a client node)” on page 1198
- “UNLOCK PROFILE (Unlock a profile)” on page 1200

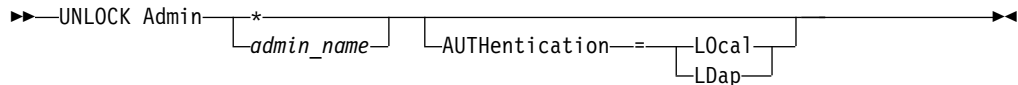
## UNLOCK ADMIN (Unlock an administrator)

Use the **UNLOCK ADMIN** command to allow a locked administrator to access the server again. You can also unlock multiple administrators that authenticate with the same method.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *admin\_name* (Required)

Specifies the name of the administrator to unlock. You can use wildcard characters to specify the administrator name. You do not have to enter an administrator name if you want to unlock all of the administrators according to their method of authentication. Use the wildcard with an authentication method to unlock multiple administrators. The parameter is required (no default wildcard).

#### **AUTHentication**

Specifies the method of password authentication that is needed for an administrator to log on.

#### **L0cal**

Specifies that you want to unlock administrator user IDs that authenticate passwords with the IBM Tivoli Storage Manager server.

#### **LDap**

Specifies that you want to unlock administrator user IDs that authenticate passwords with an LDAP directory server.

### Example: Unlock an administrator user ID

The administrator user ID JOE is locked out of Tivoli Storage Manager. Allow JOE to access the server. Issue the following command:

```
unlock admin joe
```

### Example: Unlock all administrator user IDs that authenticate passwords with an LDAP directory server

The administrator user ID that use passwords that authenticate with an LDAP directory server must be unlocked so the IDs can communicate with the Tivoli Storage Manager server.

```
unlock admin * authentication=ldap
```

## Related commands

*Table 394. Commands related to UNLOCK ADMIN*

<b>Command</b>	<b>Description</b>
LOCK ADMIN	Prevents an administrator from accessing Tivoli Storage Manager.
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.

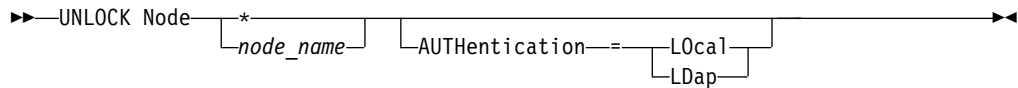
## UNLOCK NODE (Unlock a client node)

Use this command to allow a locked client node to access the server again. You can also unlock multiple nodes that use the same method of authentication.

### Privilege class

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

### Syntax



### Parameters

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

Specifies the name of the client node to unlock. You can use wildcard characters to specify the node name. You do not have to enter a node name if you want to unlock all of the nodes according to their method of authentication. Use the wildcard with an authentication method to unlock groups of nodes. The parameter is required. There is no default wildcard character available.

#### **AUTHentication**

Specifies the node password authentication method. This parameter is optional.

#### **L0cal**

Specifies that you want to unlock nodes that authenticate passwords with the IBM Tivoli Storage Manager server.

#### **LDap**

Specifies that you want to unlock nodes that authenticate passwords with an LDAP directory server.

### Example: Unlock a node

The client node SMITH is locked out of IBM Tivoli Storage Manager. Allow SMITH to access the server.

```
unlock node smith
```

### Example: Unlock all nodes that authenticate with the Tivoli Storage Manager server

The nodes that are not authenticating passwords with LDAP directory servers must be unlocked.

```
unlock node * authentication=local
```

### Related commands

Table 395. Commands related to UNLOCK NODE

Command	Description
LOCK NODE	Prevents a client from accessing the server.

*Table 395. Commands related to UNLOCK NODE (continued)*

<b>Command</b>	<b>Description</b>
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 396. 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.

- “UPDATE ADMIN (Update an administrator)” on page 1207
- “UPDATE ALERTTRIGGER (Update a defined alert trigger)” on page 1202
- “UPDATE ALERTSTATUS (Update the status of an alert)” on page 1205
- “UPDATE BACKUPSET (Update a retention value assigned to a backup set)” on page 1211
- “UPDATE CLIENTOPT (Update a client option sequence number)” on page 1216
- “UPDATE CLOPTSET (Update a client option set description)” on page 1217
- “UPDATE COLLOGROUP (Update a collocation group)” on page 1218
- “UPDATE COPYGROUP (Update a copy group)” on page 1219
- “UPDATE DATAMOVER (Update a data mover)” on page 1227
- “UPDATE DEVCLASS (Update the attributes of a device class)” on page 1229
- “UPDATE DOMAIN (Update a policy domain)” on page 1286
- “UPDATE DRIVE (Update a drive)” on page 1288
- “UPDATE FILESPACE (Update file-space node-replication rules)” on page 1292
- “UPDATE LIBRARY (Update a library)” on page 1297
- “UPDATE LIBVOLUME (Change the status of a storage volume)” on page 1313
- “UPDATE MACHINE (Update machine information)” on page 1315
- “UPDATE MGMTCLASS (Update a management class)” on page 1317
- “UPDATE NODE (Update node attributes)” on page 1320
- “UPDATE NODEGROUP (Update a node group)” on page 1338
- “UPDATE PATH (Change a path)” on page 1339
- “UPDATE POLICYSET (Update a policy set description)” on page 1347
- “UPDATE PROFILE (Update a profile description)” on page 1349
- “UPDATE RECOVERYMEDIA (Update recovery media)” on page 1350
- “UPDATE REPLRULE (Update replication rules)” on page 1352
- “UPDATE SCHEDULE (Update a schedule)” on page 1354
- “UPDATE SCRIPT (Update a Tivoli Storage Manager script)” on page 1379
- “UPDATE SERVER (Update a server defined for server-to-server communications)” on page 1382
- “UPDATE SERVERGROUP (Update a server group description)” on page 1386
- “UPDATE SPACETRIGGER (Update the space triggers)” on page 1387
- “UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)” on page 1389
- “UPDATE STGPOOL (Update a storage pool)” on page 1393
- “UPDATE VIRTUALFSMAPPING (Update a virtual file space mapping)” on page 1434
- “UPDATE VOLHISTORY (Update sequential volume history information)” on page 1436
- “UPDATE VOLUME (Change a storage pool volume)” on page 1438

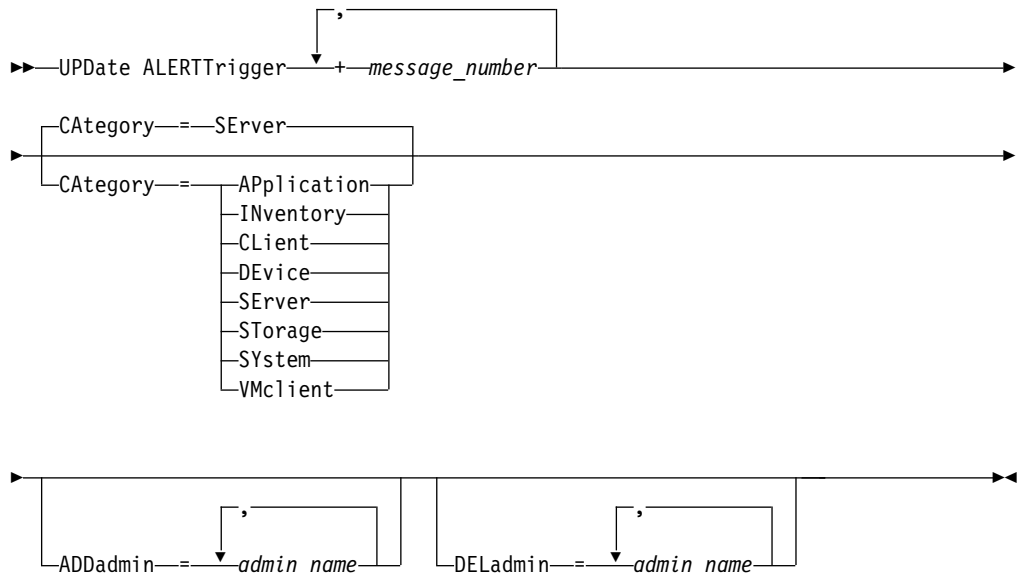
## UPDATE ALERTTRIGGER (Update a defined alert trigger)

Use this command to update the attributes of one or more alert triggers.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

#### *message\_number* (Required)

Specifies the message number that you want to associate with the alert trigger. Specify multiple message numbers, which are separated by commas, and no intervening spaces. Message numbers are a maximum of eight characters in length.

#### **CATegory**

Specifies the category type for the alert, which is determined by the message types. The default value is `SERVER`.

**Note:** Changing the category of an alert trigger does not change the category of existing alerts on the server. New alerts are categorized with the new category.

Specify one of the following values:

#### **APplication**

Alert is classified as application category. For example, you can specify this category for messages that are associated with application (TDP) clients.

#### **INventory**

Alert is classified as inventory category. For example, you can specify this category for messages that are associated with the database, active log file, or archive log file.



**Client**

Alert is classified as client category. For example, you can specify this category for messages that are associated with general client activities.

**Device**

Alert is classified as device category. For example, you can specify this category for messages that are associated with device classes, libraries, drives, or paths.

**Server**

Alert is classified as general server category. For example, you can specify this category for messages that are associated with general server activities or events.

**Storage**

Alert is classified as storage category. For example, you can specify this category for messages that are associated with storage pools.

**Systems**

Alert is classified under system clients category. For example, you can specify this category for messages that are associated with system backup and archive or hierarchical storage management (HSM) backup-archive clients.

**VMclient**

Alert is classified under VMclient category. For example, you can specify this category for messages that are associated with virtual machine clients.

**Admin**

This optional parameter specifies the name of the administrator who receives email notification of this alert. The alert trigger is defined successfully even if no administrator names are specified.

**ADDadmin**

Specifies the administrator name that you want to add to the list of administrators that receive email alerts. Specify multiple administrator names, which are separated by commas, and no intervening spaces.

**DELadmin**

Specifies the administrator name that you want to delete from the list of administrators that receive email alerts. Specify multiple administrator names, which are separated by commas, and no intervening spaces.

**Update alert trigger**

Add the names of the administrators that want to be notified when ANR1073E, ANR1074E alerts occur, and also delete the name of an administrator that no longer wants to be notified, by issuing the following command:

```
update alerttrigger ANR1073E,ANR1074E ADDadmin=djee,cdawson,mhaye deladmin=harryh
```

**Related commands**

Table 397. Commands related to **UPDATE ALERTRIGGER**

Command	Description
"DEFINE ALERTRIGGER (Define an alert trigger)" on page 124	Associates specified messages to an alert trigger.
"DELETE ALERTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.

Table 397. Commands related to UPDATE ALERTTRIGGER (continued)

Command	Description
"QUERY ALERTSTATUS (Query the status of an alert)" on page 680	Displays information about alerts that have been issued on the server.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"UPDATE ALERTSTATUS (Update the status of an alert)" on page 1205	Updates the status of a reported alert.

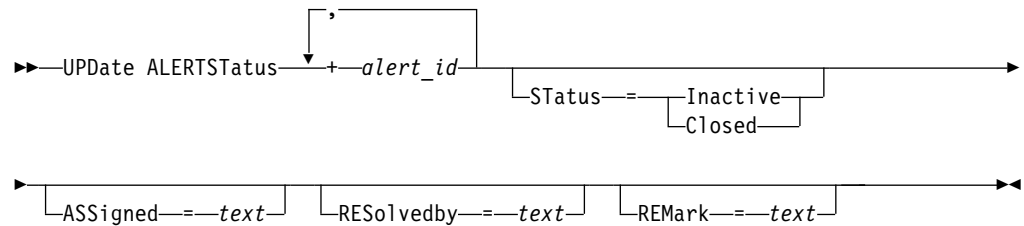
## UPDATE ALERTSTATUS (Update the status of an alert)

Use this command to update the status of a reported alert.

### Privilege class

Any administrator can issue this command.

### Syntax



### Parameters

#### *alert\_id* (Required)

Specifies the alert that you want to update. You can specify multiple message numbers by separating them with commas and no intervening spaces.

#### **Status**

Specifies the status type that you want to update. Alerts can be changed from active to inactive or closed, or from inactive to closed. Possible values are:

##### **Inactive**

Active alerts can be changed to inactive status.

##### **Closed**

Active and inactive alerts can be changed to closed status.

#### **ASSigned**

Specifies the administrator name that is assigned the alert that you want to query.

#### **RESolvedby**

Specifies the administrator name that resolved the alert that you want to query.

#### **REMark**

This parameter specifies comment text. The comment text cannot exceed 255 characters. If the description contains any blank spaces, enclose the entire text in quotation marks (""). Remove previously defined text by specifying a null string ("") for this value.

### Update the comment text in an alert

Issue the following command to update the comment text for alert ID number 25 and indicate that *DJADMIN* is working on the alert:

```
update alertstatus 25 assigned=DJADMIN
```

## Update alert status

Issue the following command to change alert ID number 72 to the closed status, and add a remark about how the alert was resolved:

```
update alertstatus 72 status=closed remark="Increased the file system size for  
the active log"
```

## Related commands

*Table 398. Commands related to UPDATE ALERTSTATUS*

Command	Description
"DEFINE ALERTTRIGGER (Define an alert trigger)" on page 124	Associates specified messages to an alert trigger.
"DELETE ALERTTRIGGER (Remove a message from an alert trigger)" on page 389	Removes a message number that can trigger an alert.
"QUERY ALERTSTATUS (Query the status of an alert)" on page 680	Displays information about alerts that have been issued on the server.
"QUERY ALERTTRIGGER (Query the list of defined alert triggers)" on page 678	Displays message numbers that trigger an alert.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
"UPDATE ALERTTRIGGER (Update a defined alert trigger)" on page 1202	Updates the attributes of one or more alert triggers.

## 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. The **SET PASSEXP** command does not affect passwords that authenticate with an LDAP directory server.

**Restriction:** You cannot update the authentication method for your own user ID. If necessary, another administrator must make that change. Also, when updating a password with the **UPDATE ADMIN** command, you cannot use a wildcard with the `admin_name` parameter.

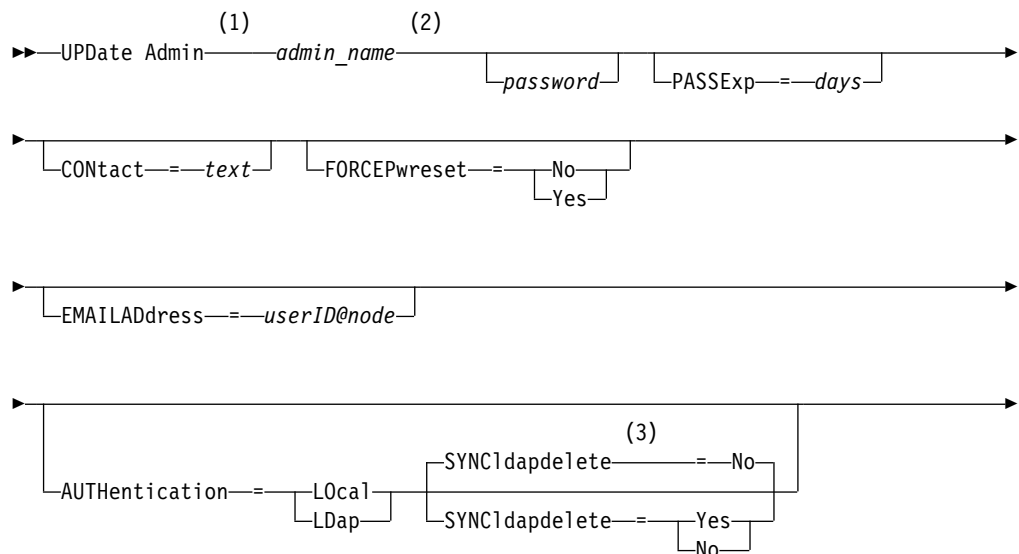
Administrators with the same name as a node can be created during a **REGISTER NODE** command. If you do not specify a name, the administrator is given the same name as the node. To keep the node and administrator with the same name synchronized, the authentication method and the **SSLREQUIRED** setting for the node are updated to match the administrator. If the administrator authentication method is changed from **LOCAL** to **LDAP** and a password is not provided, the node is put in "LDAP pending" status. A password is then requested at the next logon. Passwords between same-named nodes and administrators are kept in sync through any authentication change.

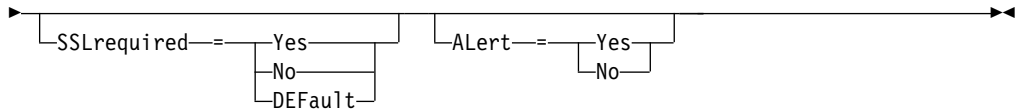
You must use the **RENAME ADMIN** command to change the name of a registered administrator.

### Privilege class

To issue this command to change another administrator ID 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.
- 2 Passwords are optional for this command, except when changing the authentication method from LDAP to LOCAL.
- 3 The **SYNC1dapdelete** parameter applies only if an administrator authenticating to an LDAP directory server reverts to local authentication.

**Parameters**

*admin\_name* **(Required)**

Specifies the name of the administrator to be updated.

*password*

Specifies the administrator's password. This parameter is optional for most cases. If the administrator authentication method is changed from LDAP to LOCAL, a password is required. 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. This parameter does not apply to passwords that are stored on an LDAP directory server.

**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 for additional contact information. The information specified by this parameter is not acted upon by Tivoli Storage Manager.

**AUTHentication**

This parameter determines the password authentication method that the administrator ID uses; either LDAP or LOCAL.

**Local**

Specifies that the administrator uses the local Tivoli Storage Manager server database to store passwords for authentication.

**Ldap**

Specifies that the administrator uses an LDAP directory server for password authentication.

**SYNCLdapdelete**

This parameter applies only if an administrator who authenticates to an LDAP directory server wants to revert to local authentication. If an entry is being used by another server and sharing a namespace in the LDAP directory server, select NO. If you want to synchronize an entry to a Tivoli Storage Manager server, and the entry is not used by another server or sharing a namespace, select YES. The default is NO.

**SSLrequired**

Specifies whether the administrator user ID must use Secure Sockets Layer (SSL) to communicate between the Tivoli Storage Manager server and the backup-archive client. When you authenticate passwords with an LDAP directory server, you must protect the sessions by using SSL or another network security method.

**Yes**

Specifies that SSL is required.

**No** Specifies that SSL is not required.

**DEFAULT**

Specifies that SSL is required for an administrator if the password that is associated with its user ID authenticates with an LDAP directory server. SSL is not required for an administrator ID that authenticates its password with the Tivoli Storage Manager server (LOCAL).

**Alert**

Specifies whether alerts are sent to an administrators email address.

**Yes**

Specifies that alerts are sent to the specified administrators email address.

**No** Specifies that alerts are not sent to the specified administrators email address. This is the default value.

**Tip:** Alert monitoring must be enabled, and email settings must be correctly defined to successfully receive alerts by email. To view the current settings, issue the **QUERY MONITORSETTINGS** command.

**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. The administrator in this example is authenticated to the Tivoli Storage Manager server. The password is not case-sensitive.

```
update admin larry secretword passexp=120
```

**Example: Change the administrator's authentication method and update the password**

Update the administrator LARRY to authenticate to an LDAP directory server and have the password SeCre#Tw0rd.

```
update admin larry SeCre#Tw0rd authentication=ldap
```

Password expiration does not apply to LDAP-authenticated passwords.

## Related commands

*Table 399. Commands related to UPDATE ADMIN*

Command	Description
QUERY ADMIN	Displays information about one or more Tivoli Storage Manager administrators.
QUERY STATUS	Displays the settings of server parameters, such as those selected by the SET commands.
"QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804	Displays information about monitoring alerts and server status settings.
REGISTER ADMIN	Defines a new administrator without granting administrative authority.
REGISTER NODE	Defines a client node to the server and sets options for that user.
RENAME ADMIN	Changes a 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 that are associated with a client node.



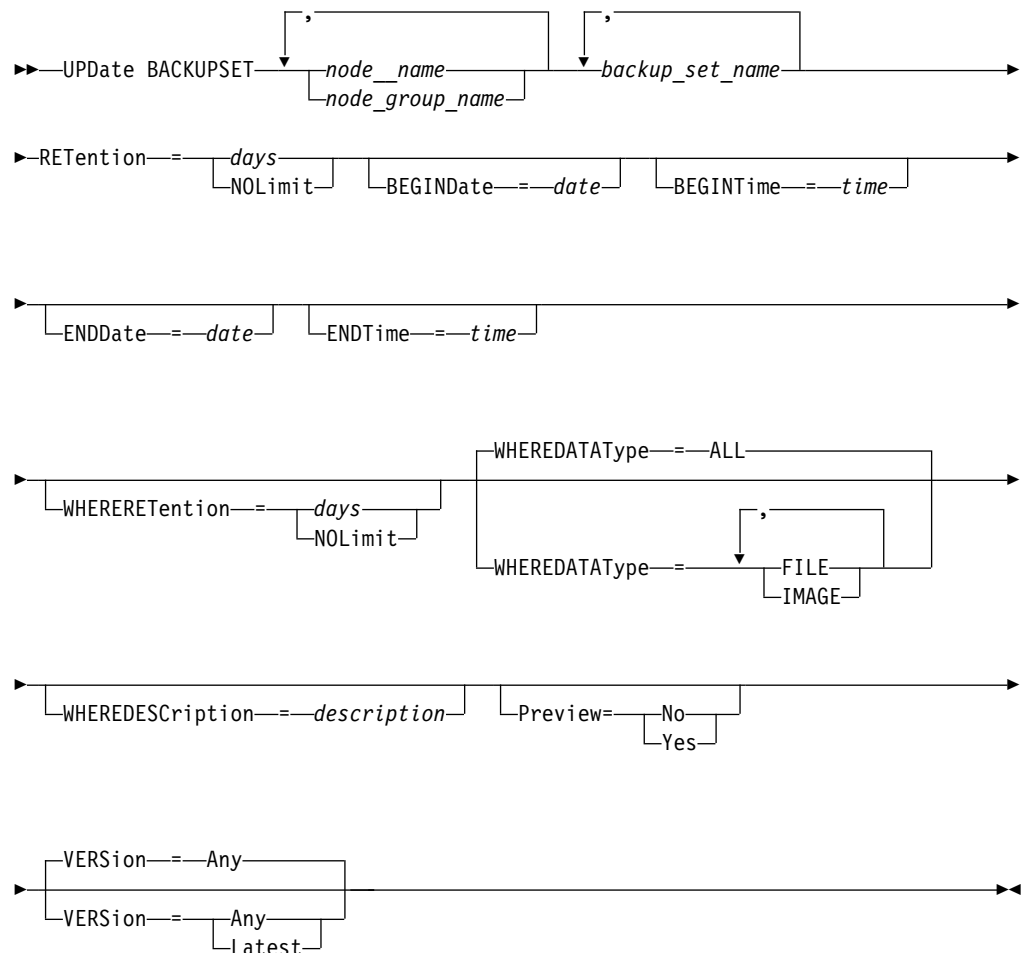
# UPDATE BACKUPSET (Update a retention value assigned to a backup set)

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
<i>TODAY+days or +days</i>	The current date plus days specified.	TODAY +3 <i>or</i> +3.
<i>TODAY-days or -days</i>	The current date minus days specified.	TODAY-3 <i>or</i> -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
<i>EOLM-days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
<i>BOTM+days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

**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> or + <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 - <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 + <i>days</i>	The current date plus days specified.	TODAY +3 or +3.
TODAY- <i>days</i> or - <i>days</i>	The current date minus days specified.	TODAY -3 or -3.
EOLM (End Of Last Month)	The last day of the previous month.	EOLM
EOLM- <i>days</i>	The last day of the previous month minus days specified.	EOLM-1  To include files that were active a day before the last day of the previous month.
BOTM (Beginning Of This Month)	The first day of the current month.	BOTM
BOTM+ <i>days</i>	The first day of the current month, plus days specified.	BOTM+9  To include files that were active on the 10th day of the current month.

### 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> or + <i>HH:MM</i>	The current time plus hours and minutes specified	NOW+02:00 or +02:00.
NOW- <i>HH:MM</i> 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.

#### **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 jane pers_data.3099
retention=70 wheredescription="healthy computers"
```

**Related commands**

*Table 400. 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 401. 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 402. 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 COLLOGROUP (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 COLLOGGroup—group_name—DESCRiption—=description—◄◄
```

### Parameters

*group\_name*

Specifies the name of the collocation group whose description you want to update.

#### DESCRiption (Required)

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, GROUPl, with a new description.

```
update colloggroup group1 "Human Resources"
```

### Related commands

Table 403. Commands related to UPDATE COLLOGROUP

Command	Description
DEFINE COLLOGROUP	Defines a collocation group.
DEFINE COLLOCMEMBER	Adds a client node or file space 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 or file space 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.
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.

- “UPDATE COPYGROUP (Update a backup copy group)” on page 1220
- “UPDATE COPYGROUP (Update a defined archive copy group)” on page 1224

*Table 404. 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.

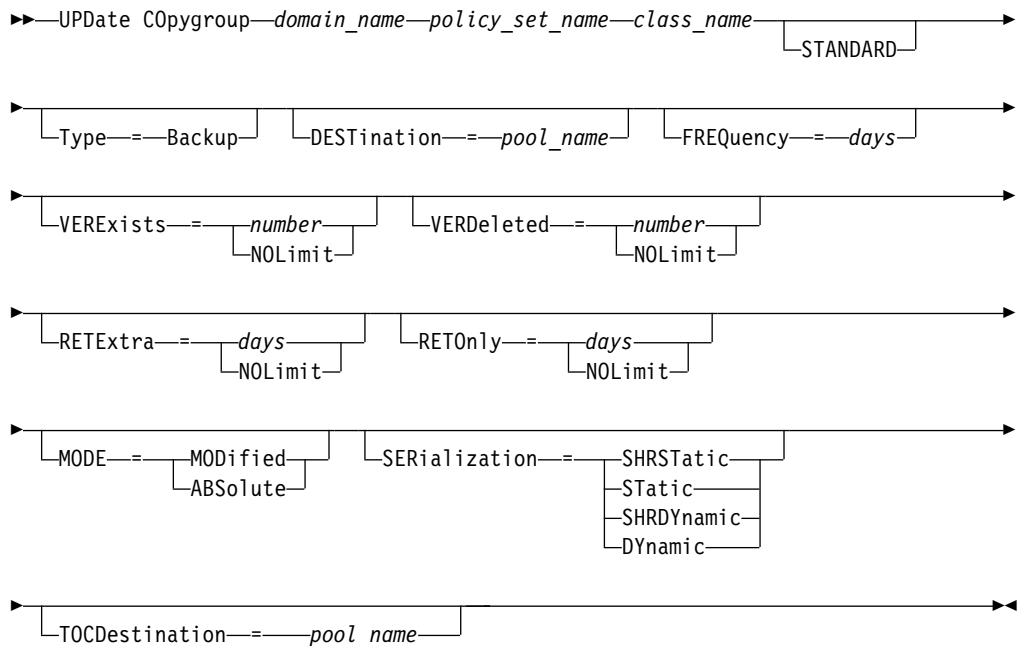
## UPDATE COPYGROUP (Update a backup copy group)

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).

### **RETOnly**

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 SHR DYNAMIC 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
```

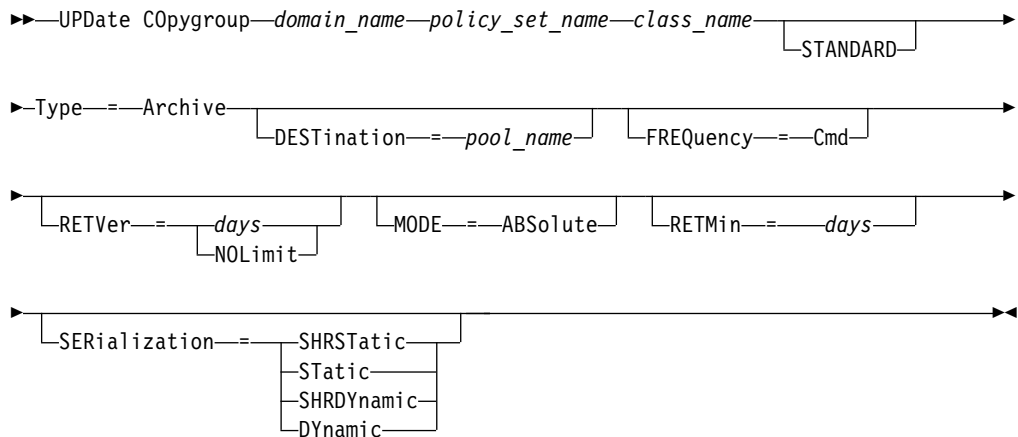
## UPDATE COPYGROUP (Update a defined archive copy group)

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** (Required)

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 **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.

**MODE=ABSolute**

Specifies that a file is always archived when the client requests it. The MODE must be ABSOLUTE. This parameter is optional.

**REMin**

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
```



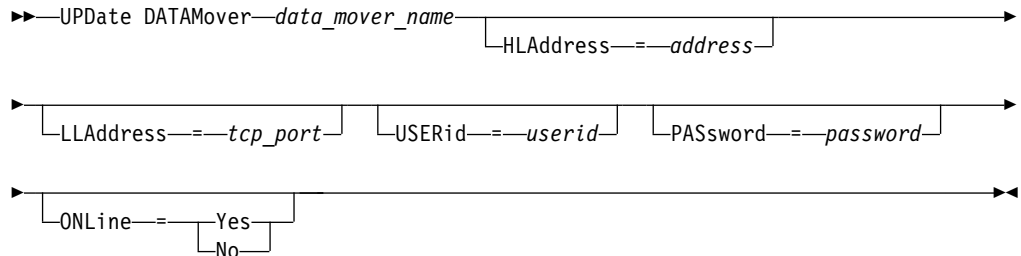
## UPDATE DATAMOVER (Update a data mover)

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 haddress=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 haddress=netapp2.tucson.ibm.com
```

## Related commands

*Table 405. Commands related to UPDATE DATAMOVER*

Command	Description
DEFINE DATAMOVER	Defines a data mover to the 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 node to the server and sets options for that user.
UPDATE NODE	Changes the attributes that are 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 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.

- “UPDATE DEVCLASS (Update a 3590 device class)” on page 1230
- “UPDATE DEVCLASS (Update a 3592 device class)” on page 1234
- “UPDATE DEVCLASS (Update a 4MM device class)” on page 1240
- “UPDATE DEVCLASS (Update an 8MM device class)” on page 1244
- “UPDATE DEVCLASS (Update a CENTERA device class)” on page 1250
- “UPDATE DEVCLASS (Update a DLT device class)” on page 1252
- “UPDATE DEVCLASS (Update an ECARTRIDGE device class)” on page 1257
- “UPDATE DEVCLASS (Update a FILE device class)” on page 1263
- “UPDATE DEVCLASS (Update a GENERICTAPE device class)” on page 1267
- “UPDATE DEVCLASS (Update an LTO device class)” on page 1269
- “UPDATE DEVCLASS (Update a NAS device class)” on page 1275
- “UPDATE DEVCLASS (Update a REMOVABLEFILE device class)” on page 1278
- “UPDATE DEVCLASS (Update a SERVER device class)” on page 1280
- “UPDATE DEVCLASS (Update a VOLSAFE device class)” on page 1282

*Table 406. Commands related to UPDATE DEVCLASS*

Command	Description
BACKUP DEVCONFIG	Backs up 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.
QUERY DEVCLASS	Displays information about device classes.
QUERY DIRSPACE	Displays information about FILE directories.
UPDATE LIBRARY	Changes the attributes of a library.

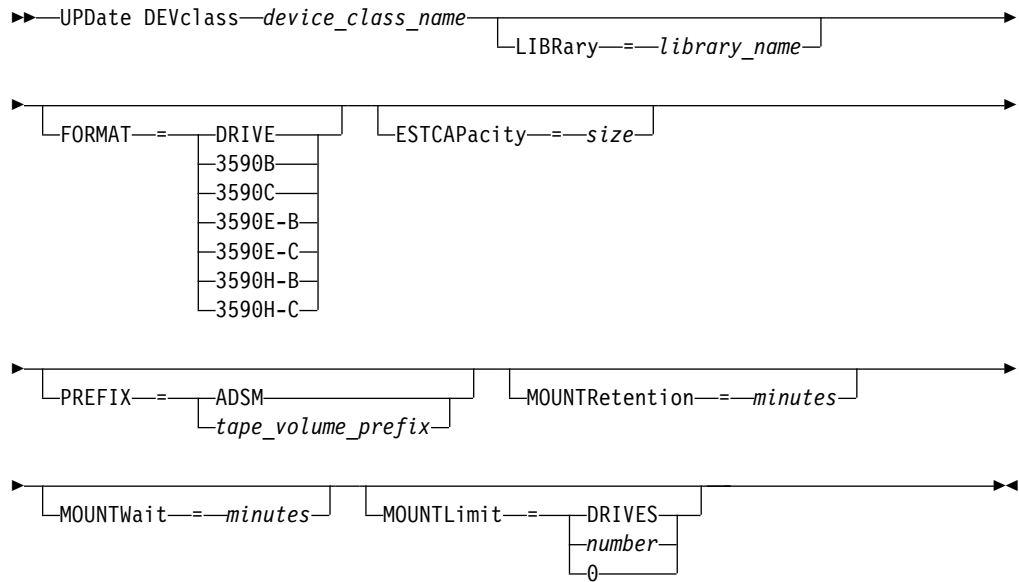
## UPDATE DEVCLASS (Update a 3590 device class)

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. 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 data is written 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.

The following tables list the recording formats, estimated capacities, and recording format options for 3590 devices:

Table 407. 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.  <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 that contains some drives that support recording formats superior to other drives.
3590B	10.0 GB	Uncompressed (basic) format
3590C	See note 20.0 GB	Compressed format
3590E-B	10.0 GB	Uncompressed (basic) format, similar to the 3590B format
3590E-C	See note 20.0 GB	Compressed format, similar to the 3590C format
3590H-B	30.0 GB (J cartridge-standard length) 60.0 GB (K cartridge-extended length)	Uncompressed (basic) format, similar to the 3590B format
3590H-C	See note 60.0 GB (J cartridge-standard length) 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 might be greater than the listed value.

Table 408. 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 that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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.

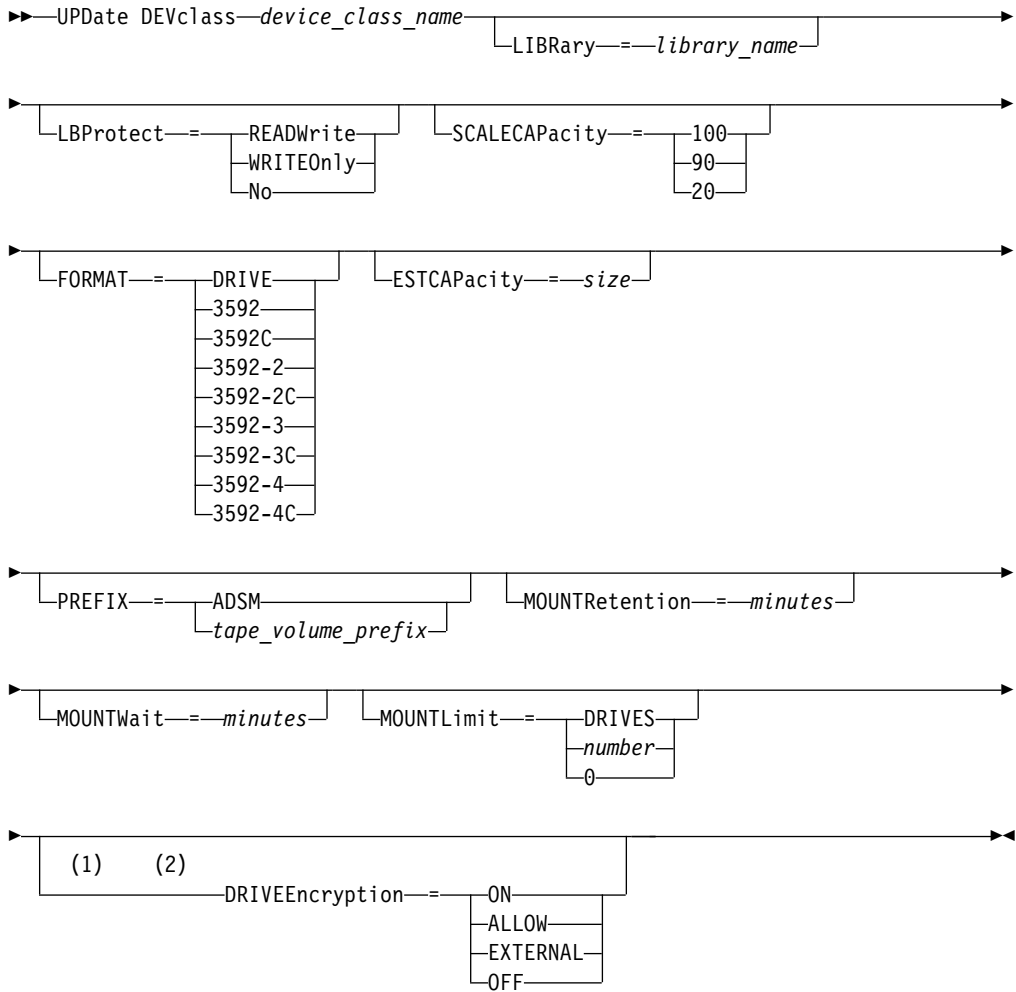
**0 (zero)**

Specifies that no new transactions can gain access to the storage pool. Any current transactions continue and complete, but new transactions are terminated.

## UPDATE DEVCLASS (Update a 3592 device class) Privilege class

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

### Syntax



### Notes:

- 1 You cannot specify both WORM=Yes and DRIVEENCRYPTION=ON.
- 2 Drive encryption is supported only for 3592 Generation 2 or later drives.

### 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. This parameter is optional.



For information about defining a library object, see the **DEFINE LIBRARY** command.

### **LBProtect**

Specifies whether logical block protection is used to ensure the integrity of data stored on tape. When **LBPROTECT** is set to **READWRITE** or to **WRITEONLY**, the server uses this feature of the tape drive for logical block protection and generates cyclic redundancy check (CRC) protection information for each data block written on tape. The server also validates the CRC protection information when data is read from the tape.

The following values are possible:

#### **READWrite**

Specifies that logical block protection is enabled in the server and the tape drive for both read and write operations. Data is stored with CRC information in each block. This mode affects performance because additional processor usage is required for Tivoli Storage Manager and the tape drive to calculate and compare CRC values. The **READWRITE** value does not affect backup sets and data that is generated by the **BACKUP DB** command.

When the **LBPROTECT** parameter is set to **READWRITE**, you do not have to specify the **CRCDATA** parameter in a storage pool definition because logical block protection provides better protection against data corruption.

#### **WRITEOnly**

Specifies that logical block protection is enabled in the server and the tape drive for write operations only. Data is stored containing CRC information in each block. For read operations, the server and the tape drive do not validate the CRC. This mode affects performance because additional processor usage is required for Tivoli Storage Manager to generate the CRC and for the tape drive to calculate and compare CRC values for write operations. The **WRITEONLY** value does not affect backup sets and data that are generated by the **BACKUP DB** command.

**No** Specifies that logical block protection is not enabled in the server and the tape drive for read and write operations. However, the server enables logical block protection on write operations for a filling volume that already has data with logical block protection.

**Restriction:** Logical block protection is supported only on IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.

See Technote 1634851 at <http://www.ibm.com/support/docview.wss?uid=swg21634851> for an explanation about when to use the **LBProtect** parameter.

### **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 takes effect when data is first written to a volume. Any updates to the device class for scale capacity do not affect volumes that already have data that is written to them until the volume is returned to scratch status.

## FORMAT

Specifies the recording format to be used when data is written 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.

The following table lists the recording formats, estimated capacities, and recording format options for 3592 devices:

Table 409. 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.  <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 that contains some drives that support recording formats superior to other drives.
3592	300 GB	Uncompressed (basic) format
3592C	See note 900 GB	Compressed format
3592-2	500 GB 700 GB	Uncompressed (basic) format JA tapes Uncompressed (basic) format JB tapes
3592-2C	1.5 TB 2.1 TB	Compressed format JA tapes Compressed format JB tapes
3592-3	640 GB 1 TB	Uncompressed (basic) format JA tapes Uncompressed (basic) format JB tapes
3592-3C	1.9 TB 3 TB	Compressed format JA tapes Compressed format JB tapes
3592-4	400 GB 1.5 TB 3.1 TB	Uncompressed (basic) format JK tapes Uncompressed (basic) format JB tapes Uncompressed (basic) format JC tape
3592-4C	1.2 TB 4.4 TB 9.4 TB	Compressed format JK tapes Compressed format JB tapes Compressed format JC tapes

**Note:** If this format uses the tape-drive hardware-compression feature, depending on the effectiveness of compression, the actual capacity might be different from the listed value.

**Important:** For optimal performance, avoid mixing different generations of drives in a single SCSI library.

Special configurations are also required for mixing different generations of 3592 drives in 349x and ACSLS libraries.

### **ESTCAPacity**

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

**DRIVEEncryption**

Specifies whether drive encryption is allowed. This parameter is optional.

Updating this parameter affects empty volumes only. If a filling volume was previously encrypted or is 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 allows drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes—for example, back up 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 is not allowed and backup operations fail.

**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is allowed if either the library or system 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 off encryption. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns off encryption.

**OFF**

Specifies that drive encryption is not allowed. If you enable either the library or system method of encryption, backups fail. If you enable the application method, Tivoli Storage Manager disables encryption and backups are attempted.

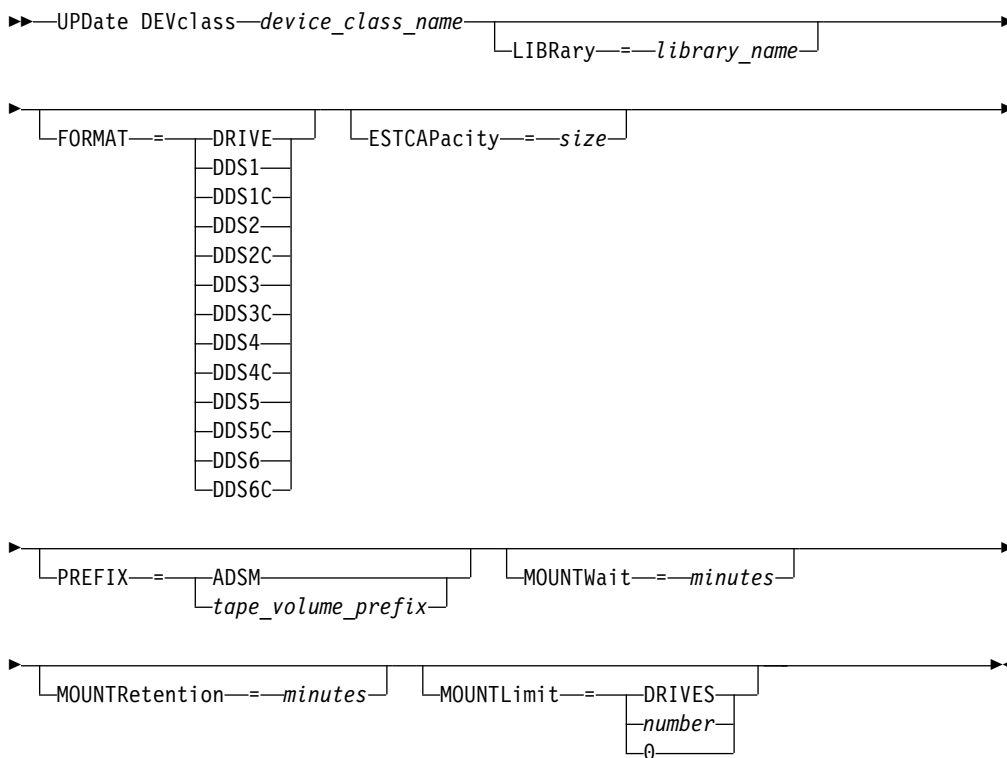
## UPDATE DEVCLASS (Update a 4MM device class)

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 data is written 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.

The following table lists the recording formats and estimated capacities for 4 mm devices:

Table 410. 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.  <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 that contains some drives that support recording formats superior to other drives.
DDS1	1.3 GB (60 meter) 2.0 GB (90 meter)	Uncompressed format, applies only to 60-meter and 90-meter tapes
DDS1C	See note 1.3 GB (60 meter) 2.0 GB (90 meter)	Compressed format, applies only to 60-meter and 90-meter tapes
DDS2	4.0 GB	Uncompressed format, applies only to 120-meter tapes
DDS2C	See note 8.0 GB	Compressed format, applies only to 120-meter tapes
DDS3	12.0 GB	Uncompressed format, applies only to 125-meter tapes
DDS3C	See note 24.0 GB	Compressed format, applies only to 125-meter tapes
DDS4	20.0 GB	Uncompressed format, applies only to 150-meter tapes
DDS4C	See note 40.0 GB	Compressed format, applies only to 150-meter tapes
DDS5	36 GB	Uncompressed format, when using DAT 72 media
DDS5C	See note 72 GB	Compressed format, when using DAT 72 media
DDS6	80 GB	Uncompressed format, when using DAT 160 media
DDS6C	See note 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 might be greater than the listed value.

### ESTCAPacity

Specifies the estimated capacity for the sequential access volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

For more information about the default estimated capacity for 4 mm tapes, see Table 410 on page 1241.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

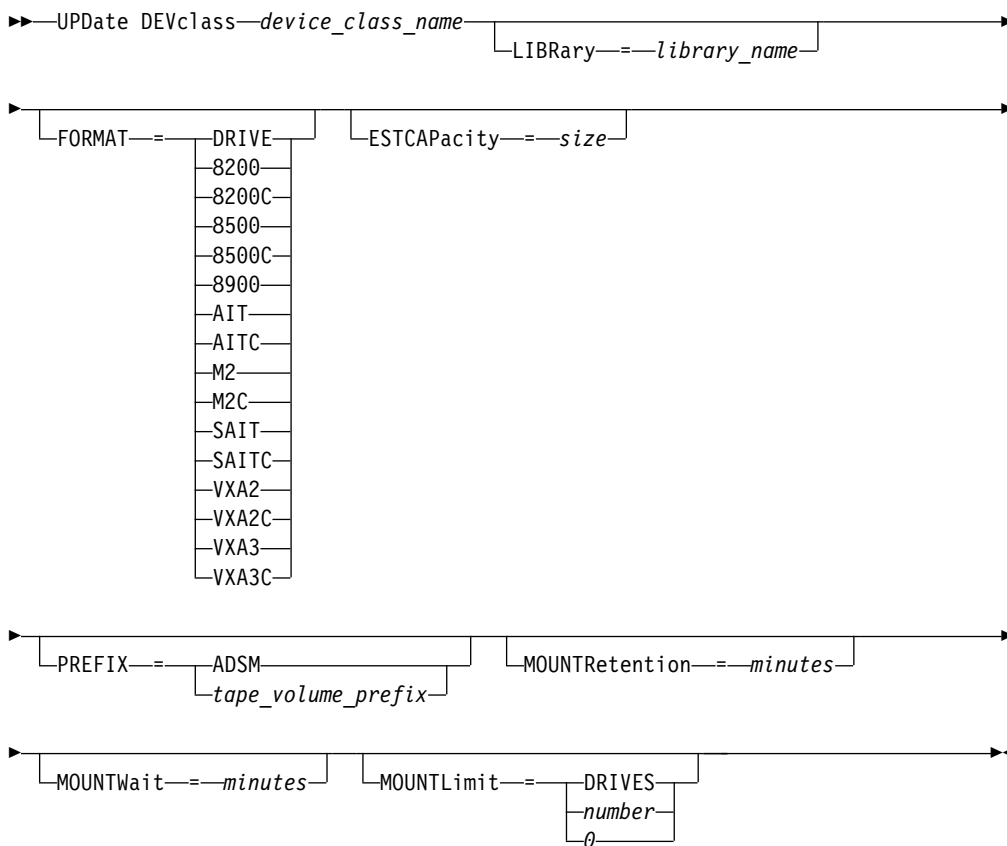
## UPDATE DEVCLASS (Update an 8MM device class)

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 more information about defining a library object, see the **DEFINE LIBRARY** command.

#### **FORMAT**

Specifies the recording format to be used when data is written 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.

The following table lists the recording formats and estimated capacities for 8 mm devices:

Table 411. 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.  <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 that contains 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 3.5 GB 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 411. 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 411. 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 might 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 that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

For more information about the default estimated capacity for 8 mm tapes, see Table 411 on page 1245.

### 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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are 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 that is named 8MMTAPE to a 15-minute mount retention.

```
update devclass 8mmtape mountretention=15
```

## UPDATE DEVCLASS (Update 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

```
▶▶ Update DEVclass device_class_name HLAddress ip_address?PEA_file (1)
└─┬──────────────────┬──────────┘ └─┬──────────┬──────────┘
  MINCAPacity size      MOUNTLimit number
```

### Notes:

- 1 For each Centera device class, you must specify an IP address. However, a Pool Entry Authorization (PEA) file name and path are optional, and the PEA file specification must follow the IP address. Use the "?" character to separate the PEA file name and path from the IP address.

### 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.

#### HLAddress

Specifies an IP address for the Centera storage device and, optionally, the name and path of one Pool Entry Authorization (PEA) file. Specify the IP address with the dotted decimal format (for example, 9.10.111.222). A Centera device might have multiple IP addresses. However, you must specify one of them as a value for this parameter.

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 that runs the Tivoli Storage Manager server. Separate the PEA file name and path from the IP address or addresses with the "?" character, for example:

```
HLADDRESS=9.10.111.222?/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 server uses the PEA file that is specified in the device class HLADDRESS parameter that was first used to open the Centera storage device.

### Note:

1. The server does not include a PEA file during installation. If you do not create a PEA file, the 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 with the command-line interface that is 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 by using the syntax `CENTERA_PEA_LOCATION=filePath_fileName`. The PEA file name and path that is specified with 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 might require a server restart if the Centera storage device identified by the IP address has already been accessed in the current instance of the server.

#### **MINCAPacity**

Specifies the new minimum size for Centera volumes that are assigned to a storage pool in this device class. This value represents the minimum amount of data that is stored on a Centera volume before the server marks it full. Centera volumes continue to accept data until the minimum amount of data is stored. This parameter is optional.

*size*

Specify this value as an integer followed by K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The minimum value that is allowed is 1 MB (`MINCAPACITY=1M`). The maximum value that is allowed is 128 GB (`MINCAPacity=128G`).

#### **MOUNTLimit**

Specifies the new maximum number of sessions that access 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 that are assigned to the same Centera device must not exceed the maximum number of sessions that are allowed by Centera.

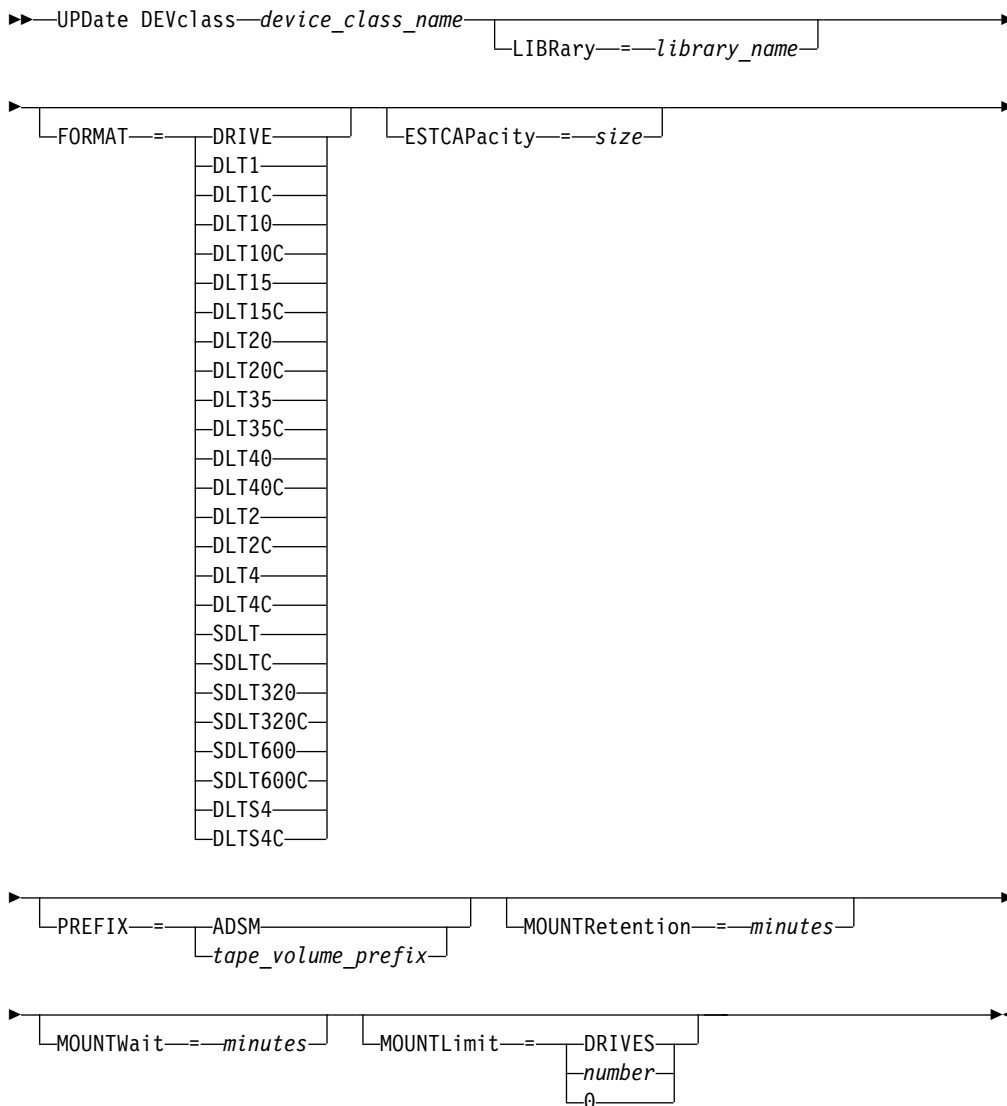
## UPDATE DEVCLASS (Update a DLT device class)

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 data is written 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.

The following table lists the recording formats and estimated capacities for DLT devices:

Table 412. 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.  <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 that contains 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 1255. 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 1255. 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) <b>Note:</b> Valid with DLT2000XT, DLT4000, and DLT7000 drives
DLT15C	See note 1 on page 1255. 30.0 GB	Compressed format, using only CompacTape IIIxt or CompacTape IV cartridges (not CompacTape III) Valid with DLT2000XT, DLT4000, and DLT7000 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 1255. 40.0 GB	Compressed format, using only CompacTape IV cartridges Valid with DLT4000, DLT7000, and DLT8000 drives
DLT35	35.0 GB	Uncompressed format, using only CompacTape IV cartridges Valid with DLT7000 and DLT8000 drives

Table 412. Recording format and default estimated capacity for DLT (continued)

Format	Estimated Capacity	Description
DLT35C	See note 1 on page 1255. 70.0 GB	Compressed format, using only CompacTape IV cartridges 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 1255. 80.0 GB	Compressed format, using CompacTape IV cartridges Valid with a DLT8000 drive
DLT2	80.0 GB	Uncompressed format, using Quantum DLT tape VS1 media
DLT2C	See note 1 on page 1255. 160.0 GB	Compressed format, using Quantum DLT tape VS1 media
DLT4	160.0 GB	Uncompressed format, using Quantum DLTtape VS1 cartridges. Valid with Quantum DLT-V4 drive
DLT4C	See note 1 on page 1255. 320.0 GB	Compressed format, using Quantum DLTtape VS1 cartridges. Valid with Quantum DLT-V4 drive
SDLT See note 2 on page 187.	100.0 GB	Uncompressed format, using Super DLT Tape 1 cartridges Valid with a Super DLT drive
SDLTC See note 2 on page 187.	See note 1 on page 187. 200.0 GB	Compressed format, using Super DLT Tape 1 cartridges Valid with a Super DLT drive
SDLT320 See note 2 on page 187.	160.0 GB	Uncompressed format, using Quantum SDLT I media Valid with a Super DLT drive
SDLT320C See note 2 on page 187.	See note 1 on page 187. 320.0 GB	Compressed format, using Quantum SDLT I media Valid with a Super DLT drive
SDLT600	300.0 GB	Uncompressed format, using SuperDLTtape-II media Valid with a Super DLT drive
SDLT600C	See note 1 on page 1255. 600.0 GB	Compressed format, using SuperDLTtape-II media Valid with a Super DLT drive
DLTS4	800 GB	Uncompressed format, using Quantum DLT S4 media. Valid with a DLT-S4 drive
DLTS4C	See note 1 on page 1255. 1.6 TB	Compressed format, using Quantum DLT S4 media. Valid with a DLT-S4 drive

Table 412. Recording format and default estimated capacity for DLT (continued)

Format	Estimated Capacity	Description
<b>Note:</b>		
1. Depending on the effectiveness of compression, the actual capacity might 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 that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

For more information about estimated capacities, see Table 412 on page 1253.

### 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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

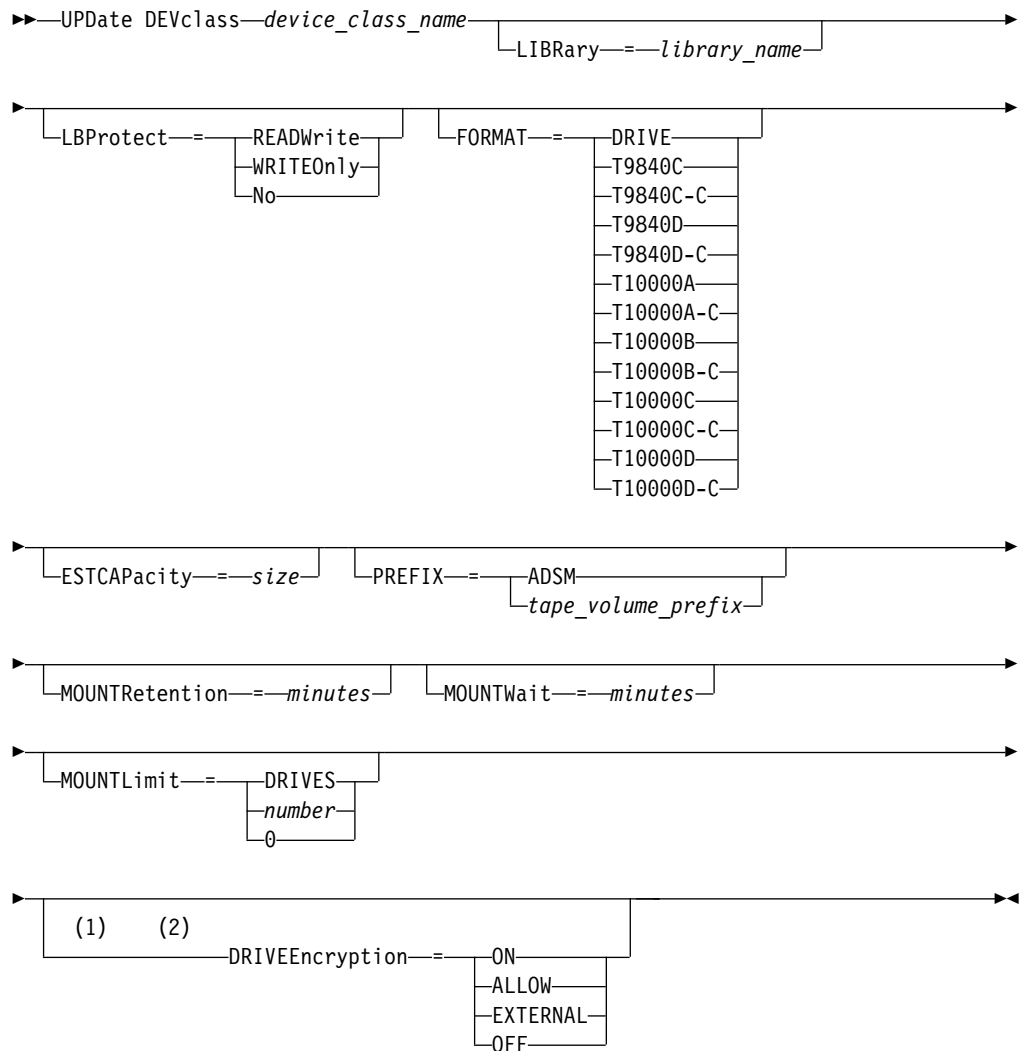
## UPDATE DEVCLASS (Update an ECARTRIDGE device class)

Use the ECARTRIDGE device class when you are using StorageTek drives such as the StorageTek T9840 or T10000.

### Privilege class

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

### Syntax



### Notes:

- 1 You can use drive encryption only for Oracle StorageTek T10000B drives with a format value of DRIVE, T10000B, or T10000B-C, for Oracle StorageTek T10000C drives with a format value of DRIVE, T10000C or T10000C-C, and for Oracle StorageTek T10000D drives with a format value of DRIVE, T10000D and T10000D-C.
- 2 You cannot specify both WORM=YES and DRIVEENCRYPTION=ON.

## 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.

### **LBProtect**

Specifies whether logical block protection is used to ensure the integrity of data stored on tape. When **LBPROTECT** is set to READWRITE or to WRITEONLY, the server uses this feature of the tape drive for logical block protection and generates cyclic redundancy check (CRC) protection information for each data block written on tape. The server also validates the CRC protection information when data is read from the tape.

The following values are possible:

#### **READWrite**

Specifies that logical block protection is enabled in the server and the tape drive for both read and write operations. Data is stored with CRC information in each block. This mode affects performance because additional processor usage is required for Tivoli Storage Manager and the tape drive to calculate and compare CRC values. The READWRITE value does not affect backup sets and data that is generated by the **BACKUP DB** command.

When the **LBPROTECT** parameter is set to READWRITE, you do not have to specify the **CRCDATA** parameter in a storage pool definition because logical block protection provides better protection against data corruption.

#### **WRITEOnly**

Specifies that logical block protection is enabled in the server and the tape drive for write operations only. Data is stored containing CRC information in each block. For read operations, the server and the tape drive do not validate the CRC. This mode affects performance because additional processor usage is required for Tivoli Storage Manager to generate the CRC and for the tape drive to calculate and compare CRC values for write operations. The WRITEONLY value does not affect backup sets and data that are generated by the **BACKUP DB** command.

**No** Specifies that logical block protection is not enabled in the server and the tape drive for read and write operations. However, the server enables logical block protection on write operations for a filling volume that already has data with logical block protection.

**Restriction:** Logical block protection is supported only on Oracle StorageTek T10000C and Oracle StorageTek T10000D drives.

### **FORMAT**

Specifies the recording format to be used when data is written 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.

**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 that is 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 413. 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.  <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 that contains some drives that support recording formats superior to other drives.
T9840C	40 GB	Uncompressed T9840C format, using a StorageTek 9840 cartridge
T9840C-C	80 GB	Compressed T9840C format, using a StorageTek 9840 cartridge
T9840D	75 GB	Uncompressed T9840D format, using a StorageTek 9840 cartridge
T9840D-C	150 GB	Compressed T9840D format, using a StorageTek 9840 cartridge
T10000A	500 GB	Uncompressed T10000A format, using a StorageTek T10000 cartridge
T10000A-C	1 TB	Compressed T10000A format, using a StorageTek T10000 cartridge
T10000B	1 TB	Uncompressed T10000B format, using an Oracle StorageTek T10000 cartridge
T10000B-C	2 TB	Compressed T10000B format, using an Oracle StorageTek T10000 cartridge
T10000C	5 TB	Uncompressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge
T10000C-C	10 TB	Compressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge
T10000D	8 TB	Uncompressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge
T10000D-C	15 TB	Compressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge

**Notes:**

- Some formats use a tape drive hardware compression feature. Depending on the effectiveness of compression, the actual capacity might 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. T10000C drives can read, but cannot write, the T10000A and T10000B formats. T10000D drives can read, but cannot write, the T10000A, T10000B, and T10000C formats.

**ESTCAPacity**

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

For more information about the default estimated capacity for cartridge tapes, see Table 413 on page 1259.

### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

**DRIVEEncryption**

Specifies whether drive encryption is allowed. This parameter is optional.

**Restriction:**

1. You can use drive encryption only for the following drives:
  - Oracle StorageTek T10000B drives that have a format value of DRIVE, T10000B, or T10000B-C
  - Oracle StorageTek T10000C drives that have a format value of DRIVE, T10000C, or T10000C-C
  - Oracle StorageTek T10000D drives that have a format value of DRIVE, T10000D, or T10000D-C
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.)
3. If encryption is enabled for a device class, and the device class is associated with a storage pool, the storage pool should not share a scratch pool with other device classes that cannot be encrypted. If a tape is encrypted, and you plan to use it on a drive that cannot be encrypted, you must manually relabel the tape before it can be used on that drive.

**ON** Specifies that Tivoli Storage Manager is the key manager for drive encryption and allows drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes are not encrypted. For example, back up sets, export volumes, and database backup volumes are not encrypted.) If you specify ON and you enable another method of encryption, drive encryption is not allowed and backup operations fail.

**ALLOW**

Specifies that Tivoli Storage Manager does not manage the keys for drive encryption. However, drive encryption for empty volumes is allowed 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 off encryption. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns off encryption.

**OFF**

Specifies that drive encryption is not allowed. If you enable another method of encryption, backups fail. If you enable the application method, Tivoli Storage Manager disables encryption and backups are attempted.

## UPDATE DEVCLASS (Update a FILE device class)

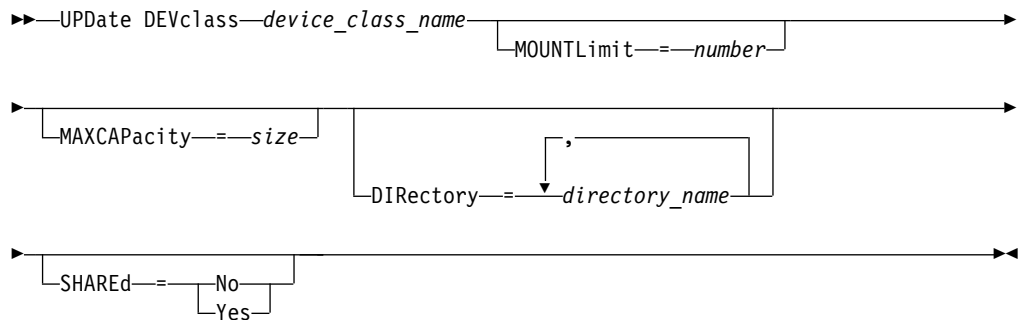
Use the FILE device class when you are using files on magnetic disk storage as volumes that store data sequentially (as on tape).

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 0 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 fails.

#### **MAXCAPacity**

Specifies the maximum size of any data storage files that are categorized by this device class. This parameter is optional.

Specify this value as an integer followed by K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The minimum size is 1 MB (**MAXCAPACITY=1M**). If you are defining a FILE device class for database-backup volumes, specify a value for MAXCAPACITY that is appropriate for the size of the database and that minimizes the number of database volumes.

For example, **MAXCAPACITY=5G** specifies that the maximum capacity for a volume in this device class is 5 gigabytes. The value that is specified must 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 that are used in this device class. Enclose the entire list of directories within quotation marks, by using commas to separate individual directory names. Special characters (for example, blank spaces) are allowed 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 the command is processed, 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 must use the DEFINE PATH command to define a path for each storage agent. The path definition includes the directory names that are used by the storage agent to access each directory.

Later, if the server must 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 that is 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 that remains in each directory. If you specify multiple directories for a device class and the directories are in the same file system, the server calculates space by adding values that represent the space that remains in each directory. These space calculations are 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 was not 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 is shared between the server and one or more storage agents. To prepare for sharing, a library is automatically defined along with a number of drives corresponding to the MOUNTLIMIT associated with the device class. If the library and drives exist and the MOUNTLIMIT is changed, drives can 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:

```
/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 stal srctype=server desttype=drive device=file  
directory="/opt/ibm1,/opt/ibm2,/opt/ibm3" library=classa
```

In this scenario, the storage agent, STA1, replaces 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.

The following results occur:

- 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 is still able to access file volume /opt/tivoli1/file1.dsm, but the storage agent STA1 is not 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 that is associated with the device class, the storage agent can lose access to a FILE volume in that directory. Although the volume is still accessible from the 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 /usr/tivoli2 and /usr/tivoli3 were specified when the device class was first defined.

```
update devclass classa
directory="/usr/tivoli2,/usr/tivoli3,/usr/otherdir"
```



## UPDATE DEVCLASS (Update a GENERICTAPE device class)

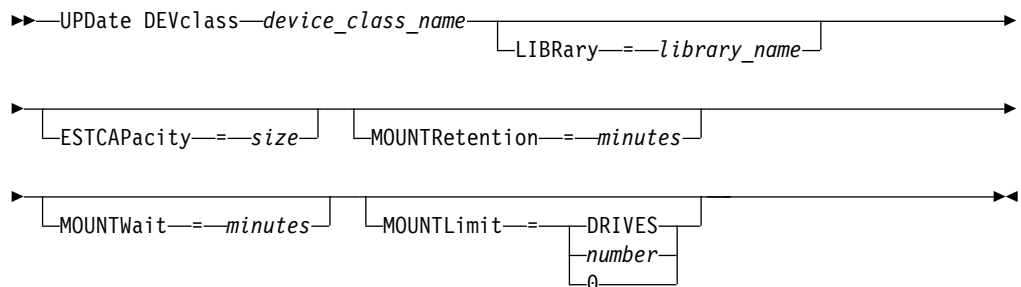
Use the GENERICTAPE device class for tape drives that are supported by operating system device drivers.

When this device type is used, 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 you define 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 updated.

#### LIBRARY

Specifies the name of the defined library object that contains the tape drives that can be 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 that are 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 that is being used.

You must specify this value as an integer followed by one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

**MOUNTRetention**

Specifies the number of minutes that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

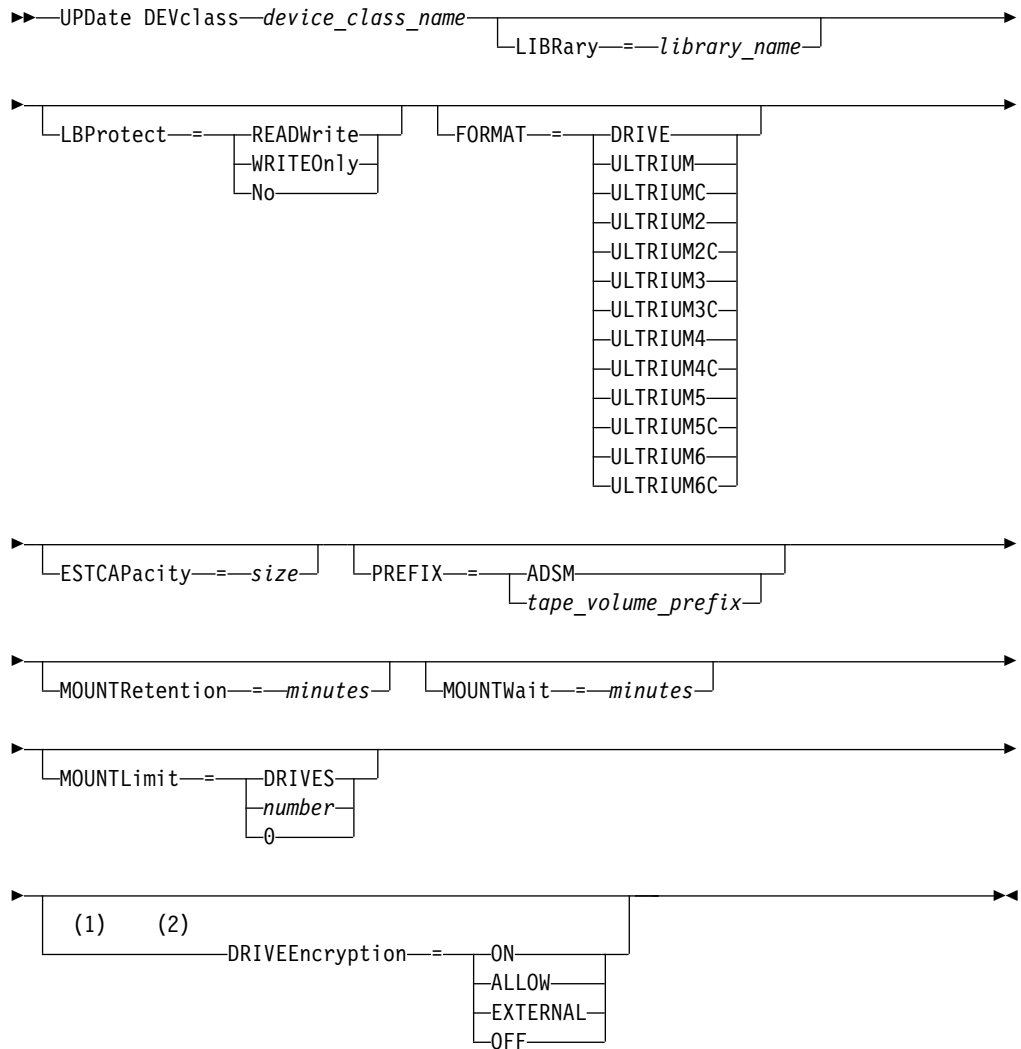
## UPDATE DEVCLASS (Update an LTO device class)

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 Ultrium 4, Ultrium 5, and Ultrium 6 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.

**LBProtect**

Specifies whether logical block protection is used to ensure the integrity of data stored on tape. When **LBPROTECT** is set to **READWRITE** or to **WRITEONLY**, the server uses this feature of the tape drive for logical block protection and generates cyclic redundancy check (CRC) protection information for each data block written on tape. The server also validates the CRC protection information when data is read from the tape.

The following values are possible:

**READWrite**

Specifies that logical block protection is enabled in the server and the tape drive for both read and write operations. Data is stored with CRC information in each block. This mode affects performance because additional processor usage is required for Tivoli Storage Manager and the tape drive to calculate and compare CRC values. The **READWRITE** value does not affect backup sets and data that is generated by the **BACKUP DB** command.

When the **LBPROTECT** parameter is set to **READWRITE**, you do not have to specify the **CRCDATA** parameter in a storage pool definition because logical block protection provides better protection against data corruption.

**WRITEOnly**

Specifies that logical block protection is enabled in the server and the tape drive for write operations only. Data is stored containing CRC information in each block. For read operations, the server and the tape drive do not validate the CRC. This mode affects performance because additional processor usage is required for Tivoli Storage Manager to generate the CRC and for the tape drive to calculate and compare CRC values for write operations. The **WRITEONLY** value does not affect backup sets and data that are generated by the **BACKUP DB** command.

**No** Specifies that logical block protection is not enabled in the server and the tape drive for read and write operations. However, the server enables logical block protection on write operations for a filling volume that already has data with logical block protection.

**Restriction:** Logical block protection is supported only on IBM LTO5 and supported LTO6 drives.

**FORMAT**

Specifies the recording format to be used when data is written 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.

When migrating all drives from Ultrium to Ultrium 2 devices:

- Delete all existing Ultrium drive definitions and the paths that are 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.

Table 414. Read - write capabilities for different generations of LTO drives.

| Drives                    | Generation 1 media | Generation 2 media | Generation 3 media | Generation 4 media | Generation 5 media | Generation 6 media |
|---------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Generation 1              | Read and write     | n/a                | n/a                | n/a                | n/a                | n/a                |
| Generation 2              | Read and write     | Read and write     | n/a                | n/a                | n/a                | n/a                |
| Generation 3 <sup>1</sup> | Read only          | Read and write     | Read and write     | n/a                | n/a                | n/a                |
| Generation 4 <sup>2</sup> | n/a                | Read only          | Read and write     | Read and write     | n/a                | n/a                |
| Generation 5 <sup>3</sup> | n/a                | n/a                | Read only          | Read and write     | Read and write     | n/a                |
| Generation 6 <sup>4</sup> | n/a                | n/a                | 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 must be checked out, and all Generation 1 storage pool volumes must be updated to read-only.

<sup>2</sup> In a library with a Generation 4 drive, all Generation 2 scratch volumes must be checked out, and all Generation 2 storage pool volumes must be updated to read-only.

<sup>3</sup> In a library with a Generation 5 drive, all Generation 3 scratch volumes must be checked out, and all Generation 3 storage pool volumes must be updated to read-only.

<sup>4</sup> In a library with a Generation 6 drive, all Generation 4 scratch volumes must be checked out, and all Generation 4 storage pool volumes must be updated to read-only.

The following table lists the recording formats and estimated capacities for LTO devices:

Table 415. 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 that contains 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                                                                                                                                                                                                                                                                                        |

Table 415. Recording format and default estimated capacity for LTO (continued).

| Format    | Estimated capacity  | Description                                                |
|-----------|---------------------|------------------------------------------------------------|
| 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              |
| ULTRIUM5  | 1.5 TB              | Uncompressed (standard) format, using Ultrium 5 cartridges |
| ULTRIUM5C | See note<br>3.0 TB  | Compressed format, using Ultrium 5 cartridges              |
| ULTRIUM6  | 2.5 TB              | Uncompressed (standard) format, using Ultrium 6 cartridges |
| ULTRIUM6C | See note<br>6.25 TB | Compressed format, using Ultrium 6 cartridges              |

**Note:** If this format uses the tape drive hardware compression feature, depending on the effectiveness of compression, the actual capacity might be greater than the listed value.

### ESTCAPacity

Specifies the estimated capacity for the sequential access volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

For more information about estimated capacities, see Table 415 on page 1271.

### 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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, 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 that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

**DRIVEEncryption**

Specifies whether drive encryption is allowed. This parameter is optional. Drive encryption is supported only for Ultrium 4, Ultrium 5, and Ultrium 6 drives and media.

**Restriction:** If encryption is enabled for a device class, and the device class is associated with a storage pool, the storage pool should not share a scratch pool with other device classes that cannot be encrypted. If a tape is encrypted, and you plan to use it on a drive that cannot be encrypted, you must manually relabel the tape before it can be used on that drive.

**ON** Specifies that Tivoli Storage Manager is the key manager for drive encryption and allows drive encryption for empty storage pool volumes only if the application method is enabled. (Other types of volumes are not encrypted. For example, back up sets, export volumes, and database backup volumes are not encrypted.) If you specify ON and you enable another method of encryption, drive encryption is not allowed and backup operations 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 allowed 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 off encryption. By contrast, when you specify ALLOW and Tivoli Storage Manager detects that AME encryption is enabled, Tivoli Storage Manager turns off encryption.

**OFF**

Specifies that drive encryption is not allowed. If you enable another method of encryption, backups fail. If you enable the application method, Tivoli Storage Manager disables encryption and backups are 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)

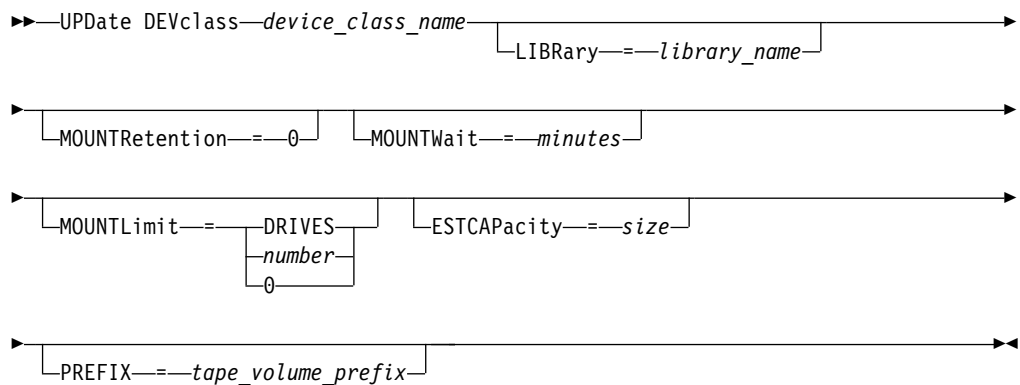
Use the NAS device class when you are using NDMP (Network Data Management Protocol) operations to back up network-attached storage (NAS) file servers. The device class is for drives that are supported by the NAS file server for backups.

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 that an idle sequential access volume is retained before it is dismounted. Zero (0) is the only supported value for device classes with DEVType=NAS.

#### MOUNTWait

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

#### **DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

#### **ESTCAPacity**

Specifies the estimated capacity for the volumes that are assigned to this device class. This parameter is optional.

You must specify this value as an integer followed by one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, national, hyphen, or numeric characters.

An example of a tape volume data set name using the default prefix is ADSM.BFS.

### **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
```

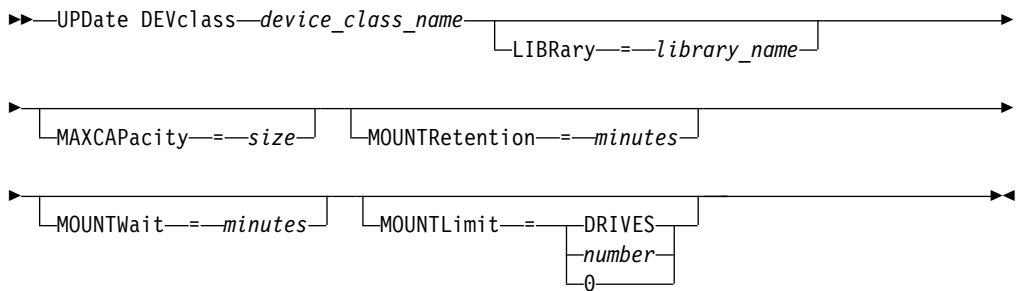
## UPDATE DEVCLASS (Update a REMOVABLEFILE device class)

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. For information about defining a library object, see the **DEFINE LIBRARY** command.

#### **MAXCAPACITY**

Specifies the maximum size of any volumes that are 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), G (gigabytes), or T (terabytes).

For example, **MAXCAPACITY=5M** specifies that the maximum capacity for a volume in this device class is 5 MB. The smallest value that is allowed is 1 MB (that is, **MAXCAPACITY=1M**).

#### **MOUNTRETENTION**

Specifies the number of minutes that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 9999.

This parameter can improve response time for sequential access media mounts by leaving previously mounted volumes online.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount

requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

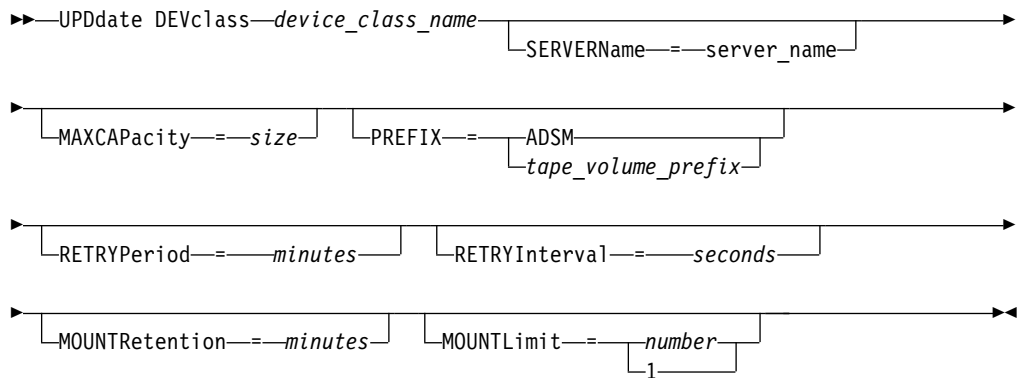
## UPDATE DEVCLASS (Update a SERVER device class)

Use the SERVER device class to use storage volumes or files that are 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** is no longer 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), G (gigabytes), or T (terabytes). The minimum value that is allowed is 1 MB (`MAXCAPACITY=1M`).

#### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:

AB.CD2.E

- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, national, hyphen, or numeric characters.

An example of a high-level archive file name that uses 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. You can specify a number 0 - 9999.

#### **RETRYInterval**

Specifies the retry interval in seconds. The retry interval is how often retries are done within a specific time period. This parameter is optional. You can specify a number 1 - 9999.

#### **MOUNTRetention**

Specifies the number of minutes to retain an idle connection with the target server before the connection is closed. This parameter is optional. You can specify a number 0 - 9999.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

#### **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 1 - 4096.

The following are possible values:

*number*

Specifies the maximum number of simultaneous sessions between the source server and the target server.

- 1 Specifies the number of simultaneous sessions between the source server and the target server.

## 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. For more information about the VolSafe device type, see “DEFINE DEVCLASS (Define a VOLSAFE device class)” on page 218.

#### FORMAT

Specifies the recording format to be used when data is written 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 that is 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 416. 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 that contains 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                                                                                                                                                                                                                                                        |
| 9840-C    | 80 GB              | LZ-1 Enhanced (4:1) compressed format, using an 80 GB cartridge with 270 meters (885 feet) of tape                                                                                                                                                                                                                                                |
| T9840C    | 40 GB              | Uncompressed T9840C format, using a StorageTek 9840 cartridge                                                                                                                                                                                                                                                                                     |
| T9840C-C  | 80 GB              | Compressed T9840C format, using a StorageTek 9840 cartridge                                                                                                                                                                                                                                                                                       |
| T9840D    | 75 GB              | Uncompressed T9840D format, using a StorageTek 9840 cartridge                                                                                                                                                                                                                                                                                     |
| T9840D-C  | 150 GB             | Compressed T9840D format, using a StorageTek 9840 cartridge                                                                                                                                                                                                                                                                                       |
| T10000A   | 500 GB             | Uncompressed T10000A format, using a StorageTek T10000 cartridge                                                                                                                                                                                                                                                                                  |
| T10000A-C | 1 TB               | Compressed T10000A format, using a StorageTek T10000 cartridge                                                                                                                                                                                                                                                                                    |
| T10000B   | 1 TB               | Uncompressed T10000B format, using an Oracle StorageTek T10000 cartridge                                                                                                                                                                                                                                                                          |
| T10000B-C | 2 TB               | Compressed T10000B format, using an Oracle StorageTek T10000 cartridge                                                                                                                                                                                                                                                                            |
| T10000C   | 5 TB               | Uncompressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge                                                                                                                                                                                                                                                                       |
| T10000C-C | 10 TB              | Compressed T10000C format, using an Oracle StorageTek T10000 T2 cartridge                                                                                                                                                                                                                                                                         |
| T10000D   | 8 TB               | Uncompressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge                                                                                                                                                                                                                                                                       |
| T10000D-C | 15 TB              | Compressed T10000D format, using an Oracle StorageTek T10000 T2 cartridge                                                                                                                                                                                                                                                                         |

### ESTCAPacity

Specifies the estimated capacity for the volumes that are 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 one of the following unit indicators: K (kilobytes), M (megabytes), G (gigabytes), or T (terabytes). The smallest value that is accepted is 1 MB (**ESTCAPACITY=1M**).

For example, specify that the estimated capacity is 9 GB with the parameter **ESTCAPACITY=9G**.

To force the Tivoli Storage Manager server to determine the estimated capacity for the volumes that are assigned to this device class, specify **ESTCAPACITY=""**.

For more information about the default estimated capacity for cartridge tapes, see Table 416 on page 1283.

### **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 a naming convention for media labels to support your current management system, use a volume prefix that conforms to your naming conventions.

Values that are specified for this parameter must meet the following conditions:

- The value is to be made up of qualifiers, which can be a maximum of eight characters including periods. For example, the following value is acceptable:  
AB.CD2.E
- The qualifiers must be separated by a single period.
- The first letter of each qualifier must be alphabetic or national (@,#,\$), followed by alphabetic, national, hyphen, or numeric characters.

An example of a high-level archive file name that uses the default prefix is ADSM.volume1.

### **MOUNTRetention**

Specifies the number of minutes that an idle sequential access volume is retained before it is dismounted. This parameter is optional. You can specify a number 0 - 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 that is managed by an external media management system), set this parameter to a low value (for example, two minutes) to enhance device sharing between applications.

**Note:** For environments in which devices are shared across storage applications, the **MOUNTRETENTION** setting must be carefully considered. This parameter determines how long an idle volume remains in a drive. Some media managers do not dismount an allocated drive to satisfy pending requests. You might need to tune this parameter to satisfy competing mount requests while maintaining optimal system performance. Typically, problems arise more frequently when the **MOUNTRETENTION** parameter is set to a value that is too small, for example, zero.

**MOUNTWait**

Specifies the maximum number of minutes the server waits 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 0 - 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 0 - 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 fails.

The following are possible values:

**DRIVES**

Specifies that every time a mount point is allocated, the number of drives that are defined and online in the library is used to calculate the true value.

**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 in this device class that are used concurrently 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 transactions continue and complete, but new transactions are terminated.

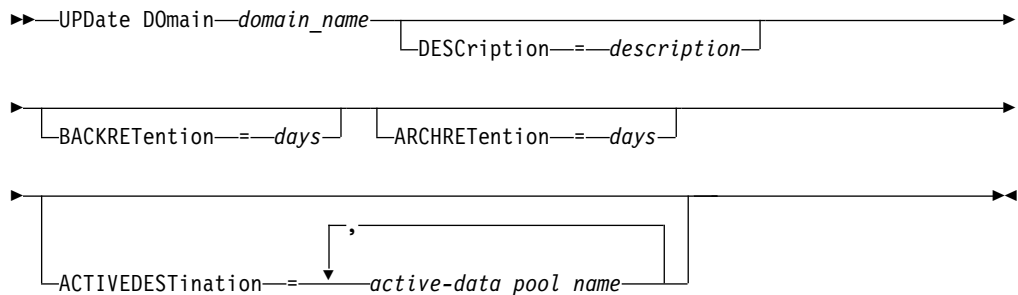
## UPDATE DOMAIN (Update a policy domain)

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 by 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 in the range 0 - 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 the new management class and the default management class do not contain a backup copy group.
- The management class to which a file is bound no longer exists. 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. 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 in the range 0 - 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. 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. 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 that are 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 10 active-data pools for a domain.

Before the Tivoli Storage Manager server writes data to an active-data pool, it verifies that the node that owns the data is assigned to a domain that has the active-data pool that is 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 simultaneous-write function is used to write data to an active-data pool, the server completes the verification during backup operations by Tivoli Storage Manager backup-archive clients or by application clients by using the Tivoli Storage Manager API. The verification is also done when active-data is being copied by using the **COPY ACTIVE**DATA command.

### Example: Update the backup retention period for a policy domain

Update the policy domain ENGPOLDOM so that the backup retention grace period is extended to 90 days and the archive retention grace period is extended to two years. Specify an active-data pool as the destination for active versions of backup data belonging to nodes that are assigned to the domain. Use *engactivedata* as the name of the active-data pool. Issue the following command:

```
update domain engpoldom description='Engineering Policy Domain'  
backretention=90 archretention=730 activedestination=engactivedata
```

### Related commands

Table 417. 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.

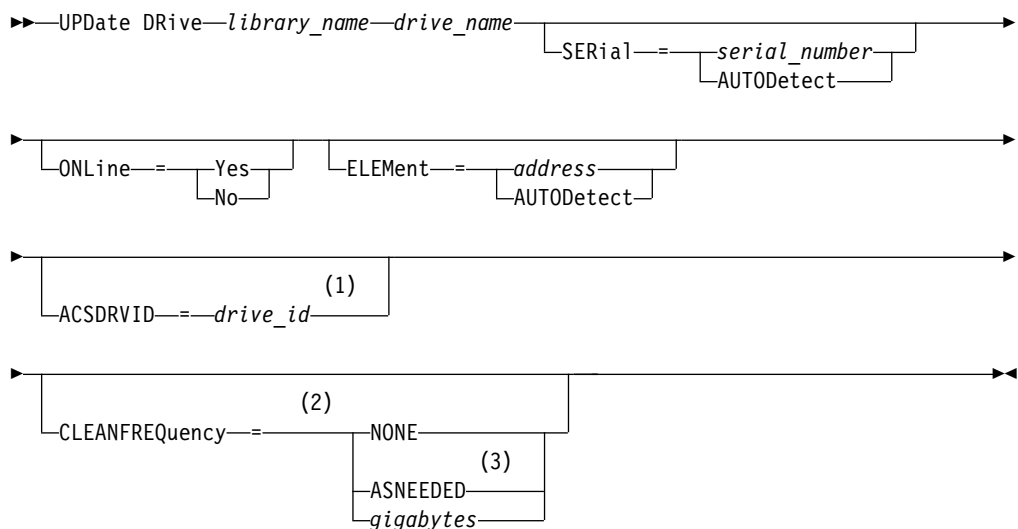
### Privilege class

For detailed and current drive support information, see the Supported Devices website for your operating system:

[http://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)

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. For more information, see the parameter description.

### 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 that are being updated. This parameter is valid only for drives in a SCSI or virtual tape library (VTL). This parameter is optional. The possible values are:

#### *serial\_number*

Specifies the serial number for the drive that is being updated.

**Note:** If a path to this drive is already defined, then the number you enter here is compared to the number detected by Tivoli Storage Manager. If the numbers do not match, the command fails.

#### **AUTODETECT**

Specifies that the serial number is automatically detected and used by Tivoli Storage Manager if a path is already defined to this drive.

If a path to this drive is not defined, then the serial number is not detected.

#### **ONLine**

Specifies whether the drive is available for use. This parameter specifies whether 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 advised. If you issue a command to take the drive offline while it is in use, an error message is issued. The mounted volume completes its current process. If this volume was part of a series of volumes for a specific transaction, the drive is not 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 is not updated, and the command fails.

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 of the drives in a library are updated to be offline, processes that need a library mount point 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 or VTL 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 only for a drive in a SCSI or VTL 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 that is being updated.

To find the element address for your library configuration, consult the information from the manufacturer.

**Remember:** If a path to this drive is already defined, then the number you enter here is compared to the number previously detected by Tivoli Storage Manager. If the numbers do not match, then this command fails.

#### **AUTODETECT**

Specifies that the element number is automatically detected and used by Tivoli Storage Manager if a path is already defined to this drive.

If a path to this drive is not defined, then the element number is not 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 fails with a Tivoli Storage Manager error message.

#### **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 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 volume inventory for the library. If you are using library based cleaning, NONE is advised when your library type supports this function. This parameter is valid only for drives in SCSI libraries, and not valid for externally managed libraries, such as 3494 libraries or StorageTek libraries that are 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.

#### **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. Visit the Supported Devices website for your operating system to view detailed drive information. If **ASNEEDED** is not supported, you can use the *gigabytes* value for automatic cleaning.

For IBM 3592 and LTO drives, library based cleaning is advised. If library based cleaning is not supported, then **ASNEEDED** must be used. *Gigabytes* is not recommended.

**Restriction:** Tivoli Storage Manager does not control the drives that are 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.

**Important:** When CLEANFREQUENCY=gigabyte, drive cleaning can occur before the gigabyte setting is reached, if the drive notifies the device driver that a cleaning is necessary.



Consult the information from the drive manufacturer 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, specify a value for the cleaning frequency to ensure that the drives receive adequate cleaning. Consult the information from the drive manufacturer for cleaning recommendations. Using the cleaning frequency that is recommended by IBM does not over clean the drives.

### Example: Update the element address for a drive

Update DRIVE3, 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, in the library named MANLIB, to take it offline.

```
update drive manlib drive3 online=no
```

### Related commands

Table 418. 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 FILESPACE (Update file-space node-replication rules)

Use this command to update file-space replication rules. You can also enable or disable replication of data to which a file space rule applies.

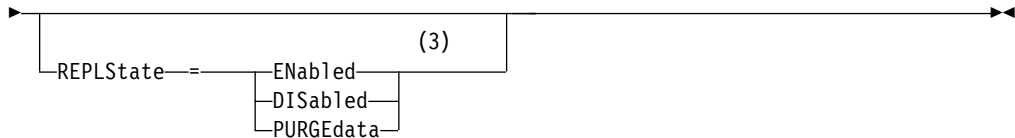
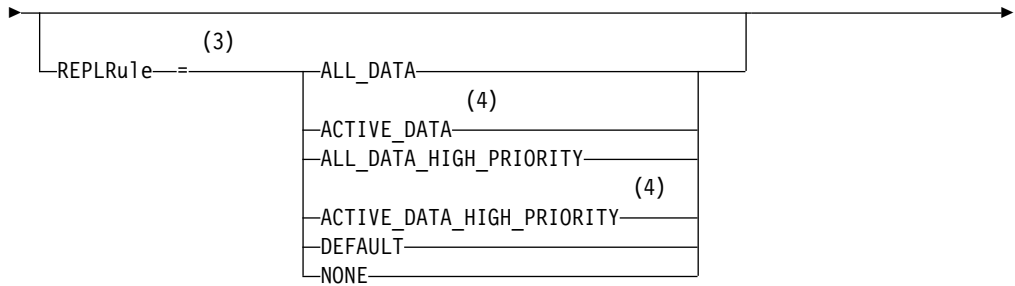
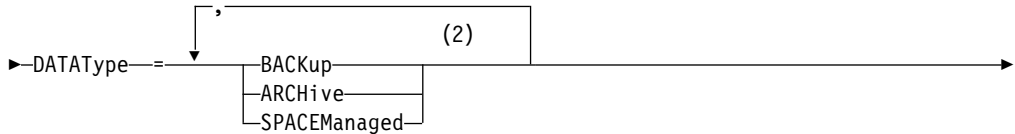
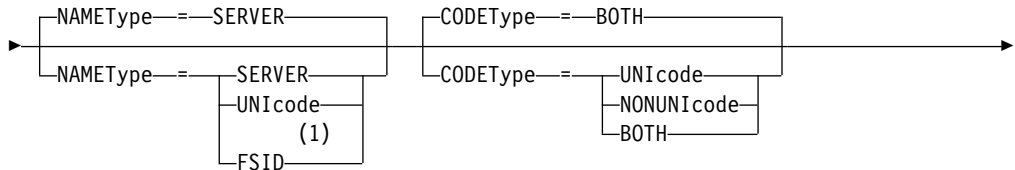
Issue this command on the server that acts as a source for replicated data.

## 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 with the file space to be updated belongs.

## Syntax

►►—UPDate Filespace—*node\_name*—*file\_space\_name*—►►



### Notes:

- 1 You cannot specify a file space identifier (FSID) if you use wildcard characters for the client node name.
- 2 You can specify each rule only once.

- 3 You must specify either the **REPLRULE** or the **REPLSTATE** parameter on this command.
- 4 The **ACTIVE\_DATA** and **ACTIVE\_DATA\_HIGH\_PRIORITY** rules are valid only if you specify **DATATYPE=BACKUP**.

## Parameters

### *node\_name* (Required)

Specifies the client node to which the file space belongs. You can use wildcard characters to specify this name. However, file space identifiers can be different among client nodes for the same file space. Therefore, you cannot specify wildcard characters for the client node name and FSID as the value for the **NAMETYPE** parameter.

### *file\_space\_name* (Required)

Specifies the name of the file space to be updated. You can use wildcard characters or a comma-delineated list to specify names.

For a server that has clients with Unicode-enabled file spaces, you might have to make the server convert the file space name that you enter. For example, you might have to make the server convert a name from the server code page to Unicode. For details, see the **NAMETYPE** parameter. If you 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. To determine the correct capitalization for the file space to be updated, use the **QUERY FILESPACE** command.

### **NAMETYPE**

Specifies how you want the server to interpret the file space names that you enter. You can use this parameter for Tivoli Storage Manager clients that Unicode-enabled and that have Windows, Macintosh OS X, or NetWare operating systems.

Use this parameter only when you enter a partly-qualified or fully-qualified file space name. The default value is **SERVER**. You can specify one of the following values:

#### **SERVER**

The server uses the server code page to interpret file space names.

#### **UNICODE**

The server converts file space names from the server code page to the UTF-8 code page. The success of the conversion depends on the operating system, on the characters in the name, and the server code page.

Conversion can fail if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion fails, the name can contain question marks, blanks, or ellipses (...).

#### **FSID**

The server interprets file space names as file space identifiers.

### **CODETYPE**

Specifies the type of file spaces to be included in node replication processing. The default value 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. You can specify one of the following values:

**UNICODE**

Specifies only file spaces that are in Unicode.

**NONUNICODE**

Specifies only file spaces that are not in Unicode.

**BOTH**

Specifies all file spaces regardless of code page type.

**DATATYPE (Required)**

Specifies the data type to which a replication rule applies. To specify multiple data types, separate the names with commas and no intervening spaces. You can specify the following values:

**BACKUP**

Specifies the backup data type.

**ARCHIVE**

Specifies the archive data type.

**SPACEMANAGED**

Specifies the space-managed data type.

**REPLRULE**

Specifies the replication rule that applies to a data type. You cannot use wildcards. If you specify multiple data types, the replication rule applies to each data type. For example, if you specify `DATATYPE=BACKUP,ARCHIVE`, the replication rule applies to backup data and to archive data.

**Restriction:** The **REPLRULE** parameter is optional. However, if you do not specify it, you must specify the **REPLSTATE** parameter.

You can specify normal-priority replication or high-priority replication rules. In a replication process that includes both normal and high-priority data, high-priority data is replicated first. Before you specify a rule, consider the order in which you want the data to be replicated.

For example, suppose that a file space contains active backup data and archive data. Replication of the active backup data is a higher priority than the archive data. To prioritize the active backup data, specify `DATATYPE=BACKUP REPLRULE=ACTIVE_DATA_HIGH_PRIORITY`. To assign a normal priority to archive data, issue the **UPDATE FILESPACE** command again, and specify `DATATYPE=ARCHIVE REPLRULE=ALL_DATA`.

You can specify the following rules:

**ALL\_DATA**

Replicates backup, archive, or space-managed data. The data is replicated with a normal priority.

**ACTIVE\_DATA**

Replicates only the active backup data in a file space. The data is replicated with a normal priority.

**Attention:** If you specify `ACTIVE_DATA` and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the `FORCERECONCILE=YES` parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

#### **ALL\_DATA\_HIGH\_PRIORITY**

Replicates backup, archive, or space-managed data. The data is replicated with a high priority.

#### **ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the `ACTIVE_DATA` replication rule except data is replicated with a high priority.

#### **DEFAULT**

Data is replicated according to the client node rule for the data type.

For example, suppose that you want to replicate the archive data in all the file spaces that belong to a client node. Replication of the archive data is a high priority. One method to accomplish this task is to specify `DATATYPE=ARCHIVE REPLRULE=DEFAULT` for each file space. Ensure that the client replication rule for archive data is set to `ALL_DATA_HIGH_PRIORITY` or to `DEFAULT`. If the client replication rule is `DEFAULT`, the server replication rule for archive data must be set to `ALL_DATA_HIGH_PRIORITY`.

#### **NONE**

Data is not replicated. For example, if you do not want to replicate the space-managed data in a file space, specify `DATATYPE=SPACEMANAGED REPLRULE=NONE`.

#### **REPLState**

Specifies the replication state for a data type. If you specified multiple data types, the state applies to all the data types. For example, if you specified `DATATYPE=BACKUP,ARCHIVE`, the state applies to backup data and archive data.

The **REPLSTATE** parameter is optional. However, if you do not specify it, you must specify the **REPLRULE** parameter. You can specify one of the following values for the **REPLSTATE** parameter:

##### **ENabled**

Specifies that the data type is ready for replication.

##### **DISabled**

Specifies that replication does not occur until you enable it.

##### **PURGEdata**

Specifies that data is deleted from the target replication server. The type of data deleted is the type of data specified by the **DATATYPE** parameter. For example, if you specify `DATATYPE=BACKUP,ARCHIVE` and

REPLSTATE=PURGEDATA, backup data and archive data are deleted from the file space on the target replication server.

After the data is deleted, the **REPLSTATE** parameter is set to DISABLED, preventing future replication of the data type or types. The replication rule for the data type is set to DEFAULT.

**Remember:** **PURGEDATA** processing does not delete file spaces. Only data is deleted. The file space shows as empty in the output of the **QUERY OCCUPANCY** command.

### Example: Update replication rules for two data types

NODE1 has three file spaces: /a, /b, and /c. The replication rules for all file spaces are set to ALL\_DATA. However, you want to replicate the backup and archive data in file space /a before the data in other file spaces is replicated.

```
update filesystem node1 /a datatype=backup,archive replrule=
  all_data_high_priority
```

### Example: Update replication rules for two data types

NODE2 has two file spaces: /a and /b. You want to temporarily suspend replication of all data in file space /b.

```
update filesystem node2 /b datatype=backup,archive,spacemanaged
  replstate=disabled
```

### Related commands

Table 419. Commands related to UPDATE FILESPACE

| Command              | Description                                                                             |
|----------------------|-----------------------------------------------------------------------------------------|
| 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 REPLICATION    | Displays information about node replication processes.                                  |
| QUERY STATUS         | Displays the settings of server parameters, such as those selected by the SET commands. |
| REPLICATE NODE       | Replicates data in file spaces that belong to a client node.                            |
| SET REPRETENTION     | Specifies the retention period for replication history records.                         |
| UPDATE NODE          | Changes the attributes that are associated with a client node.                          |
| UPDATE REPLRULE      | Enables or disables replication rules.                                                  |
| VALIDATE REPLICATION | Verifies replication for file spaces and data types.                                    |

## UPDATE LIBRARY (Update a library)

Use this command to update a library definition.

To update the device name or the external manager path name of a library, you must use the UPDATE PATH command.

Syntax and parameter descriptions are available for the following library types.

- “UPDATE LIBRARY (Update a 349X library)” on page 1298
- “UPDATE LIBRARY (Update an ACSLS library)” on page 1300
- “UPDATE LIBRARY (Update an EXTERNAL library)” on page 1302
- “UPDATE LIBRARY (Update a FILE library)” on page 1303
- “UPDATE LIBRARY (Update a manual library)” on page 1304
- “UPDATE LIBRARY (Update a SCSI library)” on page 1306
- “UPDATE LIBRARY (Update a shared library)” on page 1309
- “UPDATE LIBRARY (Update a VTL library)” on page 1310

For detailed and current library support information, see the Supported Devices website for your operating system:

[http://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)

### Related commands

Table 420. Commands related to UPDATE LIBRARY

| 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 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.                      |
| LABEL LIBVOLUME    | Labels volumes in manual or automated libraries.                    |
| 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 LIBVOLUME   | Changes the status of a storage volume.                             |
| UPDATE PATH        | Changes the attributes associated with a path.                      |

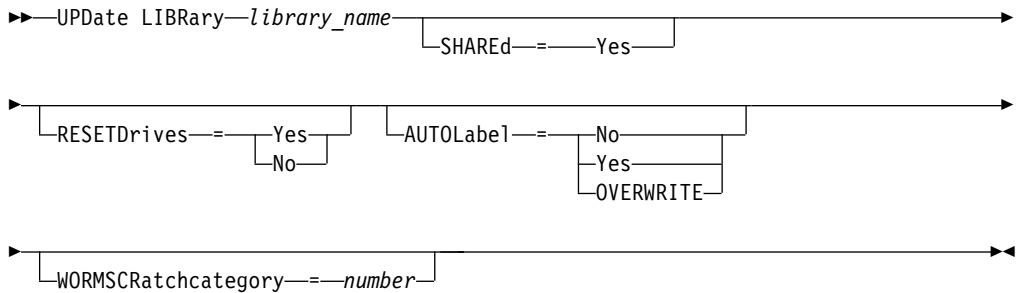
## UPDATE LIBRARY (Update a 349X library)

Use this syntax to update a 349X library.

### 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 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 server, the library cannot be shared with another server.

#### AUTOLabel

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional.

To use this option, you must 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.

#### 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 from the other category numbers in this library. This parameter is only valid when 3592 WORM volumes are used.

**Restriction:** This parameter can only be updated if the device class **WORM** parameter is set to YES and the **WORMSCRATCHCATEGORY** currently has no defined value.

**RESETDrives**

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

The following table describes the three possible configurations for drives that are attached to NAS devices.

*Table 421. Configurations for drives that are attached to NAS devices.*

| Library device configuration                                                                                                          | The behavior for persistent reserve                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The library device is attached to the Tivoli Storage Manager server, and the tape drives are shared by the server and the NAS device. | Drive reservation preemption is supported when the NAS device supports persistent reserve and it is enabled. For more information about setting persistent reserve, see the documentation for your NAS device.                         |
| The library device is attached to the Tivoli Storage Manager server and the tape drives are accessed only from the NAS device.        | Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation. |

**Yes**

Specifies that drive preemption through persistent reserve is used.

**No**

Specifies that drive preemption through persistent preserve is not used.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**Example: Add new devices to a shared library**

Update a 3494 shared library named 3494LIB2 with new device names.  
`update library 3494lib1 device=library2,library3,library4`

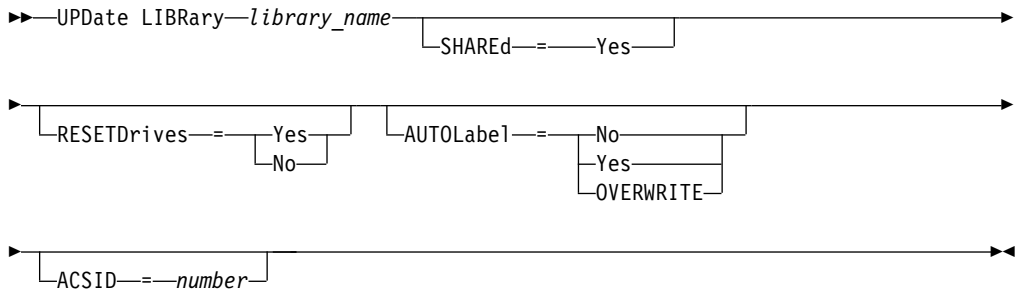
## UPDATE LIBRARY (Update an ACSLS library)

Use this syntax to update an ACSLS library.

### 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 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 server, the library cannot be shared with another server.

#### RESETDives

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

The following table describes the three possible configurations for drives that are attached to NAS devices.

Table 422. Configurations for drives that are attached to NAS devices.

| Library device configuration                                                                                                          | The behavior for persistent reserve                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The library device is attached to the Tivoli Storage Manager server, and the tape drives are shared by the server and the NAS device. | Drive reservation preemption is supported when the NAS device supports persistent reserve and it is enabled. For more information about setting persistent reserve, see the documentation for your NAS device.                         |
| The library device is attached to the Tivoli Storage Manager server and the tape drives are accessed only from the NAS device.        | Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation. |

**Yes**

Specifies that drive preemption through persistent reserve is used.

**No** Specifies that drive preemption through persistent preserve is not used.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional.

To use this option, you must 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.

**ACSID (Required)**

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.

See your StorageTek documentation for more information.

**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
```

## UPDATE LIBRARY (Update an EXTERNAL library)

Use this syntax to update an external library.

### Privilege class

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

### Syntax

```
►► Update LIBRARY library_name [AUTOLabel=No | Yes | OVERWRITE] ►►
```

### Parameters

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

Specifies the name of the library to be updated.

#### AUTOLabel

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional.

To use this option, you must 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.

### Example: Update the path name for an external library

Update an external library named EXTLIB with a new path name for the media manager.

```
update library extlib externalmanager=/v/server/mediamanager
```

## UPDATE LIBRARY (Update a FILE library)

Use this syntax to update a FILE library

### Privilege class

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

### Syntax

```
►►—UPDate LIBRARY—library_name—SHARED=      Yes—►►
```

### 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 server, the library cannot be shared with another server.

### Example: Update a FILE library to be shared

Update a file library named FILE2, so that it is shared:

```
update library file2 shared=yes
```

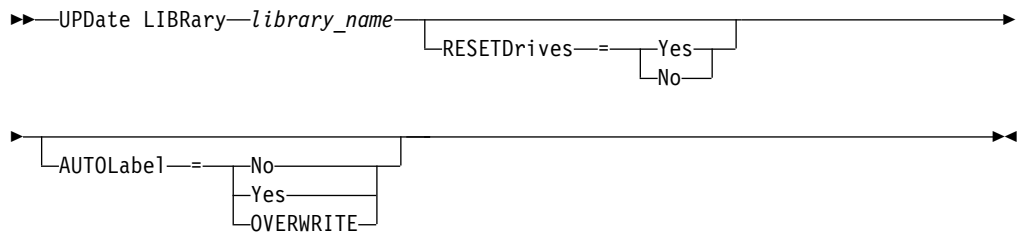
## UPDATE LIBRARY (Update a manual library)

Use this syntax to update a manual library.

### 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 be updated.

#### RESETDrives

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

#### Yes

Specifies that drive preemption through persistent reserve is used.

**No** Specifies that drive preemption through persistent preserve is not used.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

#### AUTOLabel

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional.

To use this option, you must 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 labels only 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.

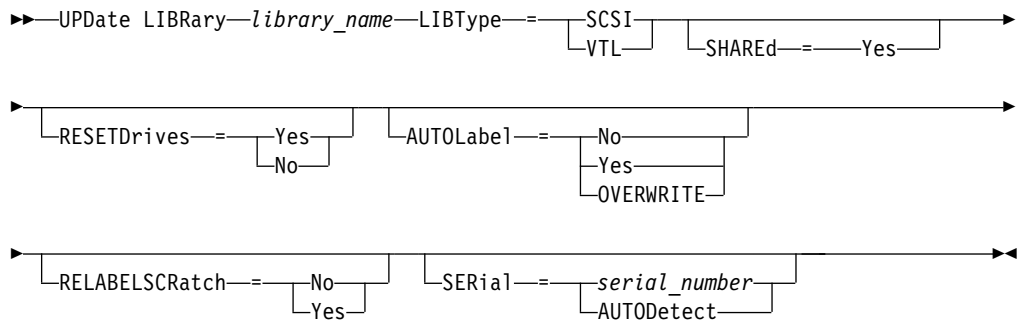
## UPDATE LIBRARY (Update a SCSI library)

Use this syntax to update a SCSI library.

### 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 be updated.

#### **LIBType** (Required)

Specifies the library type that you want to update to. Possible values are:

##### VTL

Specifies that the library has a SCSI-controlled media changer device that is represented by a Virtual Tape Library. To mount volumes on drives in this type of library, Tivoli Storage Manager uses the media changer device. This value is effective when specified for libraries with a current library type of SCSI.

**Note:** Selecting the VTL library type assumes that the following conditions are true:

- Your environment does not include mixed-media
- Paths are defined between all drives in the library and all defined servers, including storage agents, that use the library

If both conditions are not met, performance can degrade to the same levels as the SCSI library type especially during times of high stress when most drives are in use concurrently.

##### 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. This value is effective when specified for libraries with a current library type of VTL.

##### 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 server, the library cannot be shared with another server.

**RESETDrives**

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

The following table describes the three possible configurations for drives that are attached to NAS devices.

*Table 423. Configurations for drives that are attached to NAS devices.*

| Library device configuration                                                                                                                                                    | The behavior for persistent reserve                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The library device is attached to the Tivoli Storage Manager server, and the tape drives are shared by the server and the NAS device.                                           | Drive reservation preemption is supported when the NAS device supports persistent reserve and it is enabled. For more information about setting persistent reserve, see the documentation for your NAS device.                         |
| The library device is attached to the Tivoli Storage Manager server and the tape drives are accessed only from the NAS device.                                                  | Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation. |
| The library device is attached to the NAS device and accessed indirectly by NDMP (network data management protocol), and the tape drives are accessed only from the NAS device. | Drive reservation preemption is not supported. If you enable persistent reserve on the NAS device for these drives and a reservation is set by the NAS device but never cleared, you must use another method to clear the reservation. |

**Yes**

Specifies that drive preemption through persistent reserve is used.

**No**

Specifies that drive preemption through persistent preserve is not used.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

**AUTOLabel1**

Specifies whether the server attempts to automatically label tape volumes.

To use this option, you must 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.

**SERial**

Specifies the serial number for the library being updated. 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 is compared to the number detected by Tivoli Storage Manager. If the numbers do not match, the command fails. If a path has not been defined, this serial number is verified when a path is defined.

**AUTODetect**

Specifies that the serial number is 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 is not detected.

**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 are relabeled when this parameter is enabled. If the VTL includes real volumes, specifying this option might affect 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.

## UPDATE LIBRARY (Update a shared library)

Use this syntax to update a shared library.

### Privilege class

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

### Syntax

```
►►—UPDate LIBRary—library_name—PRIMarylibmanager—=—server_name—————►◄
```

### Parameters

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

Specifies the name of the library to be defined. The maximum length of this name is 30 characters.

#### PRIMarylibmanager

Specifies the name of the 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.

### 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
```

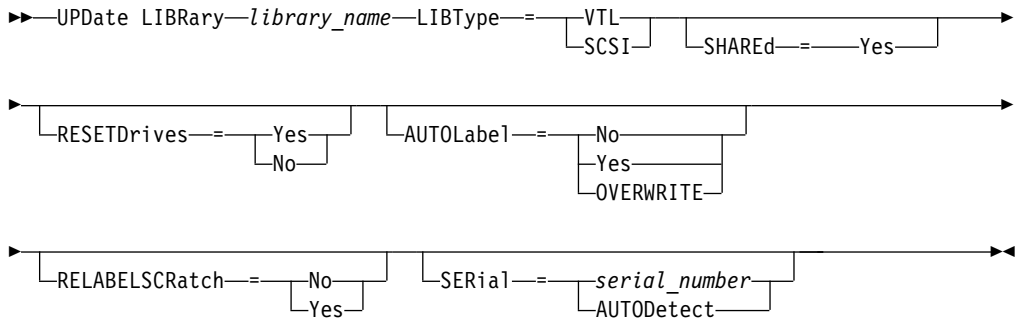
## UPDATE LIBRARY (Update a VTL library)

Use this syntax to update a library that is defined as VTL.

### 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 be defined. The maximum length of this name is 30 characters.

#### **LIBType** (Required)

Specifies the type of library that is being defined. Possible values are:

##### **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. This value is effective when specified for libraries with a current library type of VTL.

##### **VTL**

Specifies that the library has a SCSI-controlled media changer device that is represented by a Virtual Tape Library. To mount volumes on drives in this type of library, Tivoli Storage Manager uses the media changer device. This value is effective when specified for libraries with a current library type of SCSI.

**Note:** Select the VTL library type only if the following conditions are true:

- Your environment does not include mixed-media
- Paths are defined between all drives in the library and all defined servers, including storage agents, that use the library

If both conditions are not met, performance can degrade to the same levels as the SCSI library type especially during times of high stress when most drives are in use concurrently.

##### **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 server, the library cannot be shared with another server.

#### **RESETDrives**

Specifies whether the server preempts a drive reservation with persistent reserve when the server is restarted or when a library client or storage agent reconnection is established.

If persistent reserve is not supported, the server is not able to reset the path to the target device.

Support for persistent reservation has the following limitations:

- If you are using the Tivoli Storage Manager device driver, persistent reserve is only supported on some tape drives. See Technote 1470319 at <http://www.ibm.com/support/docview.wss?uid=swg21470319> for details.
- If you are using the IBM device driver, persistent reserve must be enabled at the device driver level. See the *IBM Tape Device Drivers Installation and User's Guide* at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972> for information about driver configuration.
- If you are using a virtual tape library that is emulating a supported drive, it might not support persistent reserve.

#### **Yes**

Specifies that drive preemption through persistent reserve is used.

**No** Specifies that drive preemption through persistent reserve is not used.

**Note:** A library manager will not be able to break a drive reservation if the system that has the drive reservation is not configured to use persistent reservation.

#### **AUTOLabel**

Specifies whether the server attempts to automatically label tape volumes. This parameter is optional.

To use this option, you must 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.

**Note:** If you have both virtual and real volumes in your VTL, both types are relabeled when this parameter is enabled. If the VTL includes real volumes, specifying this option might affect performance.

**Yes**

Specifies that the server relabels volumes that are deleted and returned to scratch.

**No** Specifies that the server does not relabel volumes that are deleted and returned to scratch.

**SERial**

Specifies the serial number for the library being updated. 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 is compared to the number detected by Tivoli Storage Manager. If the numbers do not match, then the command fails. If a path has not been defined, this serial number is verified when a path is defined.

**AUTODetect**

Specifies that the serial number is 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 is not detected.

## 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—►►
```

### Parameters

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

Specifies the name of the library.

#### *volume\_name* (Required)

Specifies the volume name of the storage volume.

#### STATus (Required)

Specifies a change to the status of a storage volume. Possible values are as follows:

##### **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.

**Restriction:** You cannot change the status of a volume from private to scratch if the volume belongs to a storage pool or is defined in the volume history file. You can change the status if you make a mistake when you check in volumes to the library and assign the volumes the wrong status.

### Example: Update a volume's status

Update the volume that is named WPDV00 in the library that is named AUTO to reflect a status of PRIVATE.

```
update libvolume auto wpdv00 status=private
```

### Related commands

Table 424. Commands related to UPDATE 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 424. Commands related to UPDATE LIBVOLUME (continued)

| <b>Command</b>  | <b>Description</b>                                |
|-----------------|---------------------------------------------------|
| 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.      |



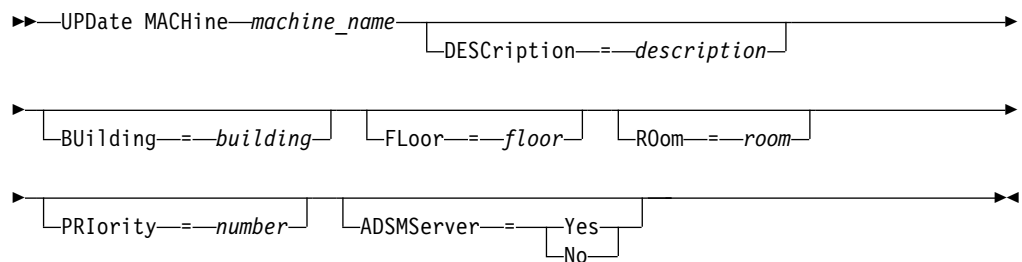
## UPDATE MACHINE (Update machine information)

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.

#### ADSM\_SERVER

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 425. Commands related to UPDATE MACHINE*

| <b>Command</b> | <b>Description</b>                                                                                 |
|----------------|----------------------------------------------------------------------------------------------------|
| DEFINE MACHINE | Defines a machine for DRM.                                                                         |
| DELETE MACHINE | Deletes a machine.                                                                                 |
| INSERT MACHINE | Inserts machine characteristics or recovery instructions into the Tivoli Storage Manager database. |
| QUERY MACHINE  | Displays information about machines.                                                               |

## UPDATE MGMTCLASS (Update a management class)

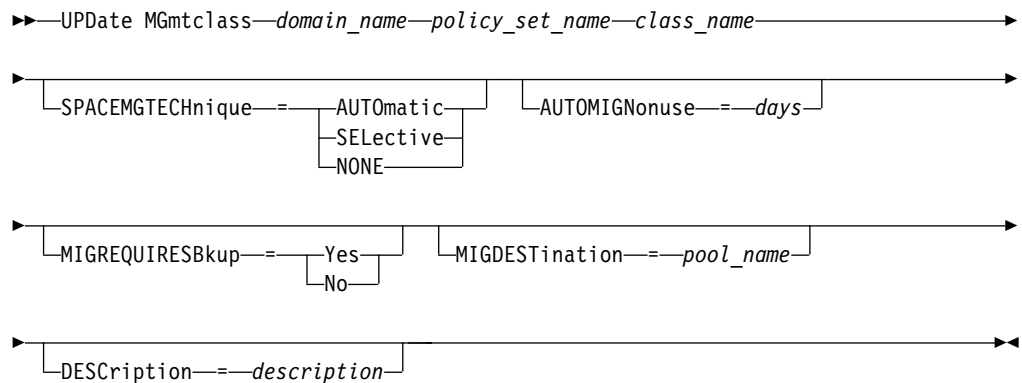
Use this command to change a management class. To allow clients to use the updated management class, you must activate the policy set that contains the management class.

**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 426. 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.                                   |
| DELETE MGMTCLASS    | Deletes a management class and its copy groups from a policy domain and policy set.        |

Table 426. Commands related to **UPDATE MGMTCLASS** (continued)

| <b>Command</b>   | <b>Description</b>                              |
|------------------|-------------------------------------------------|
| 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.

Administrators with the same name as the node are sometimes created during a **REGISTER NODE** command. If you do not specify a name, the administrator is given the same name as the node. To keep the same-named node and administrator synchronized, the authentication method and the **SSLREQUIRED** setting for the administrator ID are updated to match the node.

If you update the node authentication method or the node **SSLREQUIRED** setting and there is a same-named administrator, those administrator ID settings change.

You must have system level authority to update the node authentication method or the node **SSLREQUIRED** setting and also update a same-named administrator ID. If the same-named administrator ID has client owner authority over the node that is being updated, then system level authority is not required. You must have either unrestricted policy privilege or restricted policy privilege for the policy domain to which the client node belongs.

**Restriction:** When you update a password with the **UPDATE NODE** command, you cannot use a wildcard character with the `node_name` parameter.

When you register or update a node, you can specify whether damaged files on the node can be recovered from a target replication server. Files can be recovered only if all the following conditions are met:

- Tivoli Storage Manager, Version 7.1.1 or later, is installed on the source and target replication servers.
- The **REPLRECOVERDAMAGED** system parameter is set to ON. The system parameter can be set by using the **SET REPLRECOVERDAMAGED** command.
- The source server includes at least one file that is marked as damaged in the node that is being replicated.
- The node data was replicated before the damage occurred.

The following table describes how parameter settings affect the recovery of damaged, replicated files.

Table 427. Settings that affect the recovery of damaged files.

| Setting for the <b>REPLRECOVERDAMAGED</b> system parameter | Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REPLICATE NODE</b> command | Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REGISTER NODE</b> and <b>UPDATE NODE</b> commands | Result                                                                                                                       |
|------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| OFF                                                        | YES, NO, or not specified                                                         | YES or NO                                                                                                | During node replication, standard replication occurs and damaged files are not recovered from the target replication server. |

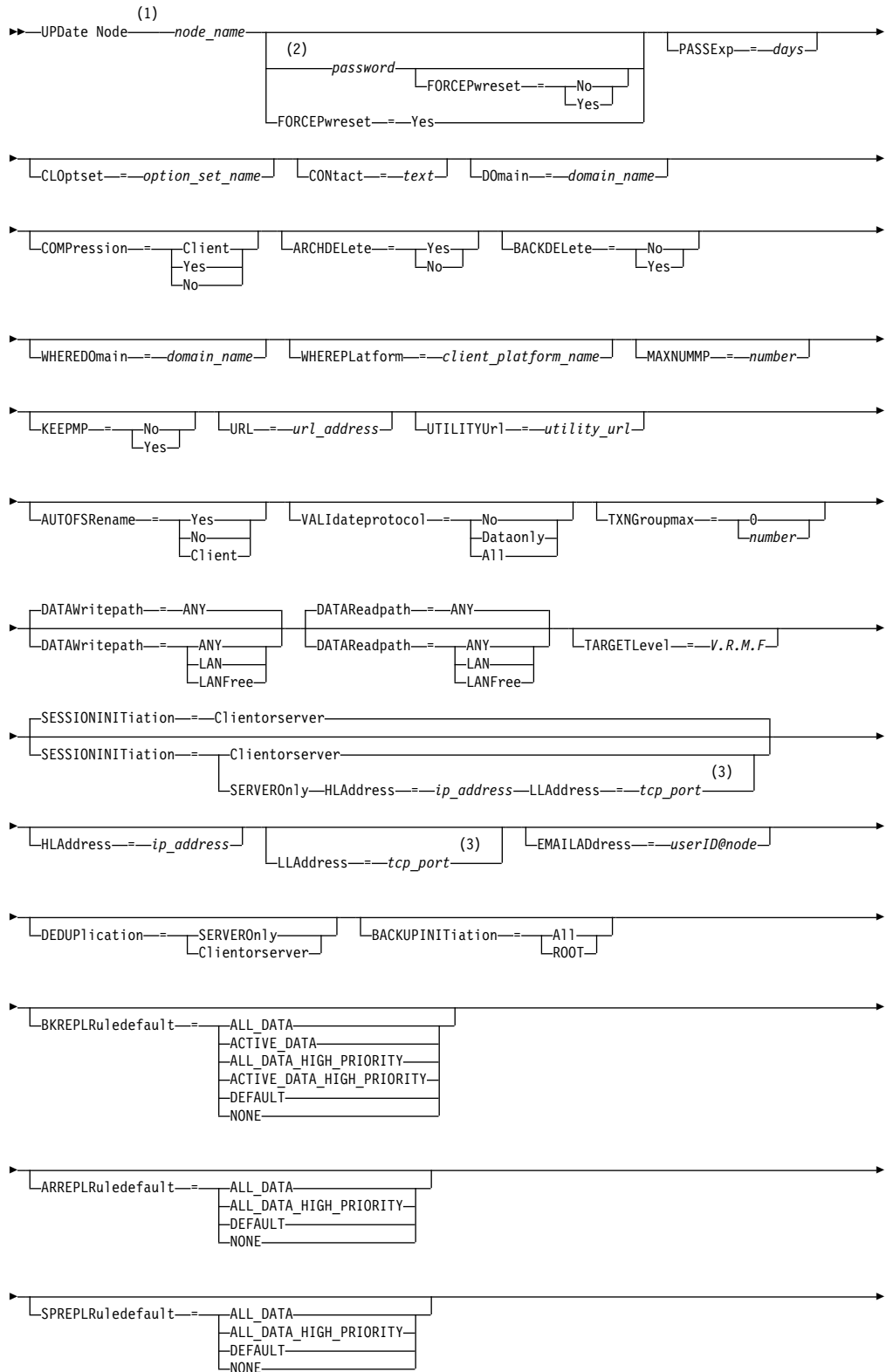
Table 427. Settings that affect the recovery of damaged files (continued).

| Setting for the <b>REPLRECOVERDAMAGED</b> system parameter | Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REPLICATE NODE</b> command | Value of the <b>RECOVERDAMAGED</b> parameter on the <b>REGISTER NODE</b> and <b>UPDATE NODE</b> commands | Result                                                                                                                             |
|------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| OFF                                                        | ONLY                                                                              | YES or NO                                                                                                | An error message is displayed because files cannot be recovered when the <b>REPLRECOVERDAMAGED</b> system parameter is set to OFF. |
| ON                                                         | YES                                                                               | YES or NO                                                                                                | During node replication, standard replication occurs and damaged files are recovered from the target replication server.           |
| ON                                                         | NO                                                                                | YES or NO                                                                                                | During node replication, standard replication occurs and damaged files are not recovered from the target replication server.       |
| ON                                                         | ONLY                                                                              | YES or NO                                                                                                | Damaged files are recovered from the target replication server, but standard node replication does not occur.                      |
| ON                                                         | Not specified                                                                     | YES                                                                                                      | During node replication, standard replication occurs and damaged files are recovered from the target replication server.           |
| ON                                                         | Not specified                                                                     | NO                                                                                                       | During node replication, standard replication occurs and damaged files are not recovered from the target replication server.       |

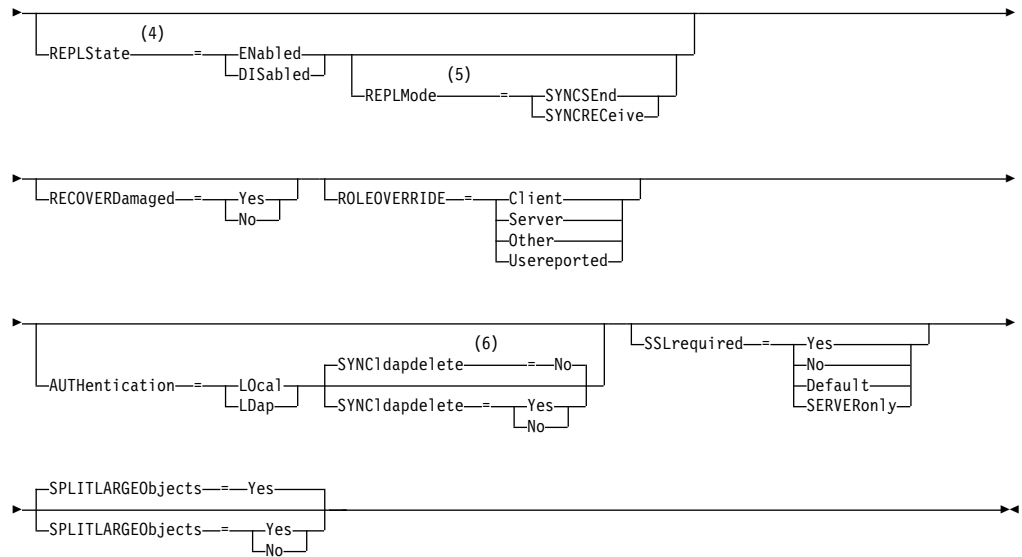
### Privilege class

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

# Syntax







**Notes:**

- 1 You must specify at least one optional parameter on this command.
- 2 Passwords are optional for this command, except when you change the authentication method from LDAP to LOCAL.
- 3 **HADDRESS** and **LADDRESS** must be previously set or specified in the **UPDATE NODE** or **REGISTER NODE** commands to use **SESSIONINITIATION=SERVERONLY**.
- 4 If you specify the **REPLSTATE** parameter and you do not specify the **REPLMODE** parameter, the replication mode of the node is set to SEND.
- 5 If you specify the **REPLMODE** parameter, you must also specify the **REPLSTATE** parameter.
- 6 The **SYNCLDAPDELETE** parameter applies only if a node that authenticates to an LDAP directory server reverts to local authentication.

**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 for most cases. If the node authentication method is changed from LDAP to LOCAL, a password is required. The maximum length of the password is 64 characters. Passwords remain current for a period that is determined by the password expiration period.

**FORCEPwreset**

Specifies whether to force a client to change or reset the password. This parameter is optional. You can specify one of the following values:

- No** Specifies that the password expiration period is set by the **SET PASSEXP** command. Do not force a client to change or reset the password while it attempts to log on to the server.

**Yes**

Specifies that the client node or administrator password will expire at the next logon. The client must change or reset the password at the next logon.

**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. To set a common expiration period for all administrators and client nodes, issue the **SET PASSEXP** command. You can also use the **SET PASSEXP** command 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 by using the **SET PASSEXP** command.

You can use the **RESET PASSEXP** command to reset the password expiration period to the common expiration period. This parameter does not apply to passwords that authenticate with an LDAP directory server.

**CLOptset**

Specifies the name of the option set to be used by the client. This parameter is optional. To remove a client option set, specify the **CLOPTSET** parameter with a null string ("").

**CONTACT**

Specifies a text string of information that identifies 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.

**Compression**

Specifies whether the client node compresses its files before it sends them to the server for backup and archive. This parameter is optional.

**Restriction:** This parameter cannot be specified for a NAS node.

You can specify one of the following values:

**Client**

Specifies that the client determines whether files are to be compressed.

**Yes**

Specifies that the client node compresses its files before it sends them to the server for backup and archive.

**No**

Specifies that the client node does not compress its files before it sends 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. You can specify one of the following values:

**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. You can specify one of the following values:

**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 might prevent the data store operations from being able to acquire mount points.

For volumes in a storage pool that is associated with the FILE or CENTERA device type, the server can have multiple sessions to read and one process to write to the same volume concurrently. To increase concurrency and provide efficient access for nodes with data in FILE or CENTERA storage pools, increase the value of the **MAXNUMMP** parameter.

For nodes that store data into primary storage pools with the simultaneous-write function that is enabled, you must adjust the value of the **MAXNUMMP** parameter to specify the correct number of mount points for each client session. A client session requires one mount point for the primary storage pool and one mount point for each copy storage pool and each active-data pool.

**URL**

Specifies the URL of the Tivoli Storage Manager web client that is configured on the client system. You can use the URL in a web browser and in the Operations Center to remotely manage the client node.

This parameter is optional. The URL must include the DNS name or IP address of the client system, and the port number that is defined on the client system for the Tivoli Storage Manager web client. For example, `http://client.mycorp.com:1581`

If you want to remove the value from this parameter, specify empty single quotation marks or empty double quotation marks with no spaces (" for single quotation marks, or "" for double quotation marks).

#### **UTILITYURL**

Specifies the address of the Tivoli Storage Manager client management services that are configured on the client system. This URL is used by the Operations Center to access client log files so that you can remotely diagnose client issues from the Operations Center.

This parameter is optional. You can specify a URL of up to 200 characters in length. The URL must start with `https`. It includes the DNS name or IP address of the client system, and the port number that is defined on the client system for the Tivoli Storage Manager client management services. For example, `https://client.mycorp.com:9028`

If you omit the port number, the Operations Center uses the port number 9028, which is the default port number when you install the client management services on the client system.

#### **KEEPMP**

Specifies whether the client node keeps the mount point for the entire session. The parameter is optional. You can specify one of the following values:

**No** Specifies that the client node releases the mount point during the session. If policy definitions cause data to be stored to a disk storage pool after data is stored 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 data is stored 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 by 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 by using the UTF-8 code page. UTF-8 is a byte-oriented encoding form that is specified by the Unicode Standard.

You can specify one of the following values:

#### **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 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).

#### **VALIDATEprotocol**

Specify whether Tivoli Storage Manager performs a cyclic redundancy check to validate the data that is sent between the client and the server. The parameter is optional. You can specify one of the following values:

**No** Specifies that Tivoli Storage Manager does not perform data validation on any data that is 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. The data does not include the file metadata. This mode affects performance because more resources are 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 affects performance because more resources are 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 might be improved by using a larger value for this option.

Specifying 0 indicates the node uses 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 increases recovery log utilization. Higher recovery log utilization might increase the risk of running out of log space. Evaluate the performance of each node before you change the parameter.

#### **DATAWritepath**

Specifies the transfer path that is 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 fails.

You can specify one of the following values:

##### **ANY**

Specifies that data is sent to the server, storage agent, or both, using any available path. A LAN-free path is used if one is available. If a LAN-free path is unavailable, the data is moved over the LAN.

##### **LAN**

Specifies that data is sent over the LAN.

##### **LANFree**

Specifies that data is sent over a LAN-free path.

#### **DATAReadpath**

Specifies the transfer path that is 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 fails. The value for the transfer path also applies to failover connections. If the value is set to LANFree, failover cannot occur for the node on the secondary server.

You can specify one of the following values:

##### **ANY**

Specifies that the server, storage agent, or both use any available path to read data. A LAN-free path is used if one is available. If a LAN-free path is unavailable, the data is read over the LAN.

##### **LAN**

Specifies that data is read over the LAN.

##### **LANFree**

Specifies that data is read by using a LAN-free path.

#### **SESSIONINITiation**

Controls whether the server or the client initiates sessions. The parameter is optional.

##### **Clientorserver**

Specifies that the client might initiate sessions with the server by communicating on the TCP/IP port that is defined with the server

option **TCPCLIENTPORT**. Server-prompted scheduling might also be used to prompt the client to connect to the server.

#### **SERVERONLY**

Specifies that the server does not accept client requests for sessions. All sessions must be initiated by server-prompted scheduling on the port that is 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 parameter must be used when **SESSIONINITIATION** is set to **SERVERONLY**, regardless of any addresses that are previously used by the client to contact the server.

The address can be specified either in numeric or host name format. If a numeric address is used, it is 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 are verified with a domain name server. Verified names are 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 parameter must be used when **SESSIONINITIATION** is set to **SERVERONLY**, regardless of any addresses that are previously used by the client to contact the server.

The value for this parameter must match the value of client option **TCPCLIENTPORT**. The default value is 1501.

#### **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 is 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 are verified with a domain name server. Verified names are 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 for more contact information. The information that is specified by this parameter is not acted upon by Tivoli Storage Manager.

**DEDUPLICATION**

Specifies where data deduplication can occur for this node. You can specify one of the following values:

**SERVEROnly**

Specifies that data stored by this node can be deduplicated on the server only.

**Clientorserver**

Specifies that data stored by this node can be deduplicated on either the client or the server. For data deduplication to take place on the client, you must also specify a value of YES for the DEDUPLICATION client option. You can specify this option in the client option file or in the client option set on the Tivoli Storage Manager server.

**TARGETLevel**

Specifies the client deployment package that is targeted for this node. You can substitute an applicable release package for V.R.M.F (Version.Release.Modification.Fix) Level. For example:  
TARGETLevel=6.2.0.0.

You must specify each segment with a number that is applicable to a deployment package. You cannot use an asterisk in any field as a substitution for a valid number. To remove an existing value, specify a null string (" "). The parameter is optional.

**Restriction:** The **TARGETLEVEL** parameter does not apply to nodes with a type of NAS or SERVER.

**BACKUPINITiation**

Specifies whether the non-root user ID on the client node can back up files to the server. The parameter is optional. The default value is ALL, indicating that non-root user IDs can back up data to the server. You can select one of the following values:

**ALL**

Specifies that non-root user IDs can back up files to the server. ALL is the default if BACKUPINITIATION is not specified.

**ROOT**

Specifies that only the root user ID can back up files to the server.

**Restriction:** The attribute is ignored by the server if the backup-archive client connects from a platform other than AIX, HP-UX, Linux, Solaris, or Mac OS.

**BKREPLRuledefault, ARREPLRuledefault, and SPREPLRuledefault**

Specifies the replication rule that applies to a data type if the file space rules for the data type are set to DEFAULT:

**BKREPLRuledefault**

Specifies the replication rule for backup data.

**ARREPLRuledefault**

Specifies the replication rule for archive data.



### **SPREPLRuledefault**

Specifies the replication rule for space-managed data.

You can specify normal-priority replication or high-priority replication rules. In a replication process that includes both normal and high-priority data, high-priority data is replicated first. Before you specify a rule, consider the order in which you want the data to be replicated.

For example, suppose that a client node contains active backup data and archive data. Replication of the active backup data is a higher priority than the archive data. To prioritize both types of data, specify `BKREPLRULEDEFAULT=ACTIVE_DATA_HIGH_PRIORITY ARREPLRULEDEFAULT=ALL_DATA`.

You can specify the following rules:

#### **ALL\_DATA**

Replicates active and inactive backup data, archive data, or space-managed data. The data is replicated with a normal priority.

#### **ACTIVE\_DATA**

Replicates only active backup data. The data is replicated with a normal priority. This rule is valid only for **BKREPLRULEDEFAULT**.

**Attention:** If you specify `ACTIVE_DATA` and one or more of the following conditions are true, inactive backup data on the target replication server is deleted, and inactive backup data on the source replication server is not replicated.

- When a Tivoli Storage Manager version earlier than Version 7.1.1 is installed on either the source or target replication servers.
- When you are using the **REPLICATE NODE** command with the `FORCERECONCILE=YES` parameter.
- When you are running the initial replication of a file space after you configure replication, restore the database, or upgrade both the source and target replication servers from a Tivoli Storage Manager version earlier than V7.1.1.

If the previous conditions are not true, all new and changed files since the last replication are replicated, including inactive files, and files are deleted when they expire.

#### **ALL\_DATA\_HIGH\_PRIORITY**

Replicates active and inactive backup data, archive data, or space-managed data. Data is replicated with a high priority.

#### **ACTIVE\_DATA\_HIGH\_PRIORITY**

This rule is the same as the `ACTIVE_DATA` replication rule except data is replicated with a high priority. This rule is valid only for **BKREPLRULEDEFAULT**.

#### **DEFAULT**

Replicates data according to the server replication rule for backup data.

For example, suppose that you want to replicate the archive data in all the file spaces that belongs to a client node. Replication of the archive data is a high priority. One method to accomplish this task is to specify `ARREPLRULEDEFAULT=DEFAULT`. Ensure that the file space rules for archive data are also set to `DEFAULT` and that the server rule for archive data is set to `ALL_DATA_HIGH_PRIORITY`.

**Restriction:** If a node is configured for replication, the file space rules are set to DEFAULT after the node stores data on the source replication server.

#### **NONE**

Data of the specified type is not replicated.

For example, if you do not want to replicate space-managed data that belongs to a client node, specify `SPREPLRULEDEFAULT=NONE`

#### **REPLState**

Specifies whether data that belongs to the client node is ready to be replicated. This parameter is optional. You can specify one of the following values:

##### **ENabled**

Specifies that the client node is ready for replication.

##### **DISabled**

Specifies that replication does not occur until you enable it.

The system response to these settings depends on the following factors:

#### **Whether the client node definition exists only on the source replication server and you are configuring the client node for replication for the first time**

If you set the replication state to **ENABLED** or **DISABLED**, the replication mode of the node on the source replication server is automatically set to **SEND** after the **UPDATE NODE** command is issued. When replication first occurs, a client node definition on the target server is automatically created. The replication state of the client node on the target server is automatically set to **ENABLED**. The replication mode is set to **RECEIVE**.

#### **Whether the client node definition exists on the source and the target replication servers, and the node data was previously replicated**

For replication to occur, the replication state of the client node on both the source and the target servers must be set to **ENABLED**. For example, if the replication state of a client node on the source server is **ENABLED** and the replication state on the target server is **DISABLED**, replication does not occur.

#### **Whether the client node definition exists on the source and the target replication servers, and the node data was previously exported from the source replication server and imported to the target replication server**

In this case, you are configuring the client nodes to synchronize the data between the two servers. When replication first occurs, the replication state of the client node on the target server is automatically set to **ENABLED**. Data on the source and target servers is synchronized.

**Restriction:** To synchronize data, you must specify the **REPLMODE** parameter in addition to the **REPLSTATE** parameter.

You can specify the **REPLMODE** parameter only if the client node has never been replicated:

- If the client node definition exists only on the source replication server, the replication mode of the node on the source replication server is automatically set to **SEND** when the **UPDATE NODE** command is issued. The replication mode of the node on the target replication server is automatically set to **RECEIVE**.

- If data that belongs to the node was previously replicated, the replication mode of the node on the source replication server is SEND. The replication mode of the node on the target replication server is RECEIVE.

### **REPLMode**

Specifies whether to synchronize the data that belongs to this client node. Specify this parameter only if data that belongs to the client node was exported from the source replication server and imported to the target replication server. Synchronization occurs during replication.

To synchronize data, you must issue the **UPDATE NODE** command on both the source and target replication servers and specify the **REPLMODE** and **REPLSTATE** parameters. The value that you specify for the **REPLMODE** parameter depends on whether the server is a source of or a target for replicated data.

You can specify one of the following values:

#### **SYNCSEnd**

Specifies that data that belongs to this client node is synchronized with data on a target server during replication. Specify this value only on the server that exported the data. When the synchronization is complete, the replication mode for the client node on the source server is automatically set to SEND. The replication mode remains SEND unless you remove the node by issuing the **REMOVE REPLNODE** command.

#### **SYNCRECeive**

Specifies that data that belongs to this client node is synchronized with data on a source server during replication. Specify this value only on the server that imported the data. When the synchronization is complete, the replication mode for the client node on the target server is automatically set to RECEIVE. The replication mode remains RECEIVE unless you remove the node by issuing the **REMOVE REPLNODE** command.

### **Restrictions:**

- You can set the **REPLMODE** parameter only if the initial replication state is NONE. To synchronize data, you change the replication state to ENABLED or DISABLED and specify a value for the **REPLMODE** parameter.
- Data can be synchronized only if you specified DATES=ABSOLUTE on the **IMPORT NODE** command. If you specified DATES=RELATIVE to import data, you must rename the node or delete its data before replication. If you do not take one of these steps, you can lose data.
- If the **REPLMODE** parameter was set incorrectly, you must issue the **REMOVE REPLNODE** command before you update the client node definition. For example, suppose that you updated the definition of a client node whose data you wanted to replicate. The data that belongs to the node was previously exported to the target replication server. You specified ENABLED as the setting of the **REPLSTATE** parameter. However, you did not specify SYNCSEND on the source replication server. As a result, the **REPLMODE** parameter was automatically set to SEND, and data that belongs to the node could not be synchronized or replicated.

Issuing **REMOVE REPLNODE** sets the replication state and the replication mode to NONE. After the **REMOVE REPLNODE** command is completed, reissue the **UPDATE NODE** command with the correct parameters and values.

### **RECOVERDamaged**

Specifies whether damaged files can be recovered for this node from a target replication server. The parameter is optional. The default value is YES. You can specify one of the following values:

**Yes**

Specifies that recovery of damaged files from a target replication server is enabled for this node.

**No** Specifies that recovery of damaged files from a target replication server is not enabled for this node.

**Tip:** The value of the **RECOVERDAMAGED** parameter is only one of several settings that determine whether damaged files are recovered. For information about how to specify the settings, see Settings that affect the recovery of damaged files.

**ROLEOVERRIDE**

Specifies whether to override the reported role of the client for processor value unit (PVU) estimation reporting. The default is **USERREPORTED**.

The role reported by the client is either client-device (for example, a workstation) or server-device (for example, file/print server, application server, database). By default, the client reports its role that is based on the client type and the operating system. All clients initially report their role as server-device, except for Tivoli Storage Manager backup-archive clients that are running Microsoft Windows workstation distributions (Windows Vista) and Macintosh OS X.

Specify one of the following values:

**Client**

Specifies a client-device.

**Server**

Specifies a server-device.

**Other**

Specifies that this node is not to be used for PVU estimation reporting. The Other value is useful when there are multiple nodes that are deployed for a physical system (for example, virtual environments, test nodes, retired nodes, and nodes not in production or clustering).

**Usereported**

Use the reported role that is provided by the client.

**AUTHentication**

This parameter determines the password authentication method that you use; either LDAP or LOCAL.

**Local**

Specifies that the node uses the local Tivoli Storage Manager server database to store passwords.

**LDap**

Specifies that the node uses an LDAP directory server to authenticate passwords. Passwords are not stored in the Tivoli Storage Manager database.

**SYNC1dapdelete**

This parameter applies only if you want a node that authenticates with an LDAP directory server, to change to authenticate with the Tivoli Storage Manager. The default is **NO**.

**Yes**

Select **YES** if you want to synchronize an entry on the Tivoli Storage

Manager server, and the entry is not used by another server and not sharing a namespace on the LDAP directory server.

**Remember:** When you select YES you update all other servers that are sharing namespace, which keeps those nodes in sync.

**No** Select NO if an entry is being used by another server and shares a namespace on the LDAP directory server.

#### **SSLrequired**

Specifies whether the node must use Secure Sockets Layer (SSL) to communicate with the Tivoli Storage Manager server. The parameter is optional. When you authenticate passwords with an LDAP directory server, you must protect the sessions by using SSL or another network security method.

#### **Yes**

Specifies that SSL is required.

**No** Specifies that SSL is not required.

#### **DEFault**

Specifies that SSL is required for a node if its password authenticates with an LDAP directory server. SSL is not required for a node that authenticates its password with the Tivoli Storage Manager server (LOCAL).

#### **SERVERonly**

Specifies that SSL is required between the backup-archive client and the server. SSL is not required between the backup-archive client and the storage agent.

#### **SPLITLARGEObjects**

Specifies whether large objects that are stored by this node are automatically split into smaller pieces, by the server, to optimize server processing. Specifying Yes causes the server to split large objects (over 10 GB) into smaller pieces when stored by a client node. Specifying No bypasses this process. Specify No only if your primary concern is maximizing throughput of backups directly to tape. The default value is Yes.

### **Example: Update node SIMON to authenticate with an LDAP directory server and connect using SSL**

```
update node simon authentication=ldap sslrequired=yes
```

Specifying **SSLREQUIRED** does not configure the server for SSL. You must follow the instructions for connecting with SSL in order for the example to work.

### **Example: Update a node with software release information for a future deployment**

The client deployment feature helps you update a backup-archive client to a newer release. The information that is generated from the **UPDATE NODE** command can help you when you plan a deployment. The information is stored for a future deployment and can be viewed by issuing the **QUERY NODE** command. After a deployment, you can issue the **QUERY NODE** command to see the current level and the target level. For example, to update node LARRY to backup-archive client Version 6.3.0.0.

```
update node LARRY targetlevel=6.3.0.0
```

### **Example: Update a node backup to compress data and keep the client from deleting archived files**

Update node LARRY so that the data on node LARRY is 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 value to 1,000.

```
update node larry txngroupmax=1000
```

### **Example: Update a node and allow it to deduplicate on the client**

Update a node BOB so that it can deduplicate on the client.

```
update node bob deduplication=clientorserver
```

### **Example: Update the role of node BOB to a server-device for PVU estimation reporting**

If you want to accumulate PVU values, only server device roles are recorded. You can update a node from client-device to server-device by issuing the **UPDATE NODE** command. For this example, node BOB is updated to a server-device.

```
update node bob role=server
```

### **Example: Update a node definition on a source replication server**

NODE1 is defined to a source replication server. The data that belongs to NODE1 was previously exported to a target replication server. Update the replication rule for backup data that belongs to NODE1 so that active backup data is replicated with a high priority. Enable replication for the node. Set up data synchronization with the target replication server.

```
update node node1 bkreplruledefault=active_data_high_priority  
replstate=enabled replmode=syncsend
```

### **Example: Update a node definition to enable recovery of damaged files**

Update the PAYROLL node to enable the recovery of damaged files from a target replication server.

```
update node payroll recoverdamaged=yes
```

## **Related commands**

*Table 428. Commands related to UPDATE NODE*

| <b>Command</b>    | <b>Description</b>                                                           |
|-------------------|------------------------------------------------------------------------------|
| 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 PVUESTIMATE | Displays an estimate of the client-devices and server-devices being managed. |

Table 428. Commands related to **UPDATE NODE** (continued)

| <b>Command</b>             | <b>Description</b>                                                                                       |
|----------------------------|----------------------------------------------------------------------------------------------------------|
| QUERY REPLNODE             | Displays information about the replication status of a client node.                                      |
| REGISTER ADMIN             | Defines a new administrator without granting administrative authority.                                   |
| REGISTER NODE              | Defines a client node 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.                         |
| REMOVE REPLNODE            | Removes a node from replication.                                                                         |
| RENAME NODE                | Changes the name for a client node.                                                                      |
| REPLICATE NODE             | Replicates data in file spaces that belong to a client node.                                             |
| RESET PASSEXP              | Resets the password expiration for nodes or administrators.                                              |
| SET DEDUPVERIFICATIONLEVEL | Specifies the percentage of extents verified by the server during client-side deduplication.             |
| SET PASSEXP                | Specifies the number of days after which a password is expired and must be changed.                      |
| SET REPLRECOVERDAMAGED     | Specifies whether node replication is enabled to recover damaged files from a target replication server. |
| UPDATE ADMIN               | Changes the password or contact information associated with any administrator.                           |
| UPDATE FILESPACE           | Changes file-space node-replication rules.                                                               |

## 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 (Required)

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 429. 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.

Syntax and parameter descriptions are available for the following path types.

- “UPDATE PATH (Change a path when the destination is a drive)” on page 1340
- “UPDATE PATH (Change a path when the destination is a library)” on page 1344

For detailed and current device support information, see the Supported Devices website for your operating system:

[http://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)

### Related commands

*Table 430. Commands related to UPDATE PATH*

| Command          | Description                                                         |
|------------------|---------------------------------------------------------------------|
| DEFINE DATAMOVER | Defines a data mover to the 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.                            |

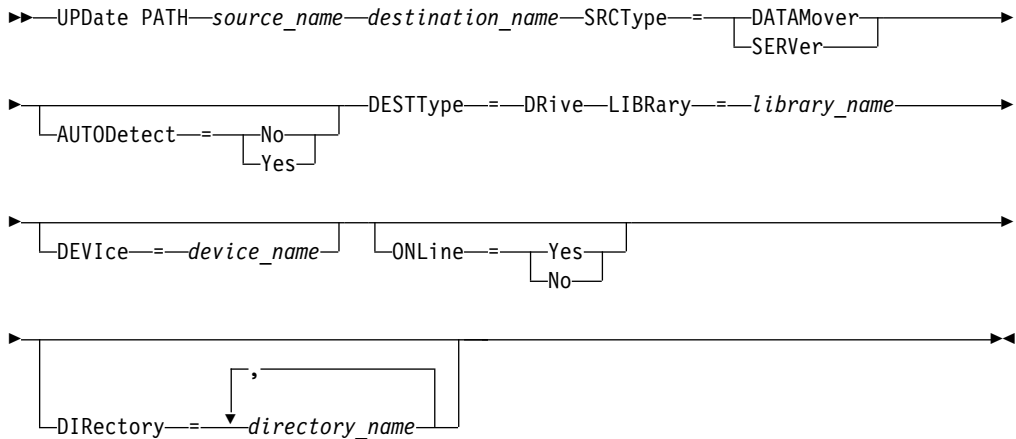
## UPDATE PATH (Change a path when the destination is a drive)

Use this syntax when updating a path definition to a drive.

### Privilege class

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

### Syntax



### 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.

#### 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 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 is not automatically updated.

##### Yes

Specifies that the serial number is automatically updated to reflect the same serial number that the drive reports to 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. If you set DESTTYPE=DRIVE and AUTODETECT=YES, then the drive element number in the 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=DRive (Required)**

Specifies that a drive is the destination. When the destination is a drive, you must specify a library name. This parameter is required.

**LIBRARY**

Specifies the name of the library to which the drive is assigned. The library and its drives must already be defined to the 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. See Table 431 for examples.

*Table 431. Examples of device names*

| Source to destination                                                        | Example                                                                                                                 |
|------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Server to a drive (not a FILE drive)                                         | /dev/rmt/tsmmt3                                                                                                         |
| 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 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 information about the device name when the source is a storage agent, see the Tivoli Storage Manager for Storage Area Networks product information.
- For 349X libraries, the alias name is a symbolic name that is specified in the /etc/ibmatl.conf file. For more information, see *IBM Tape Device Drivers Installation and User's Guide*, which can be downloaded from the IBM Systems support site at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972>.
- 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

### **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.

### **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 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
```

```
/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 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
```

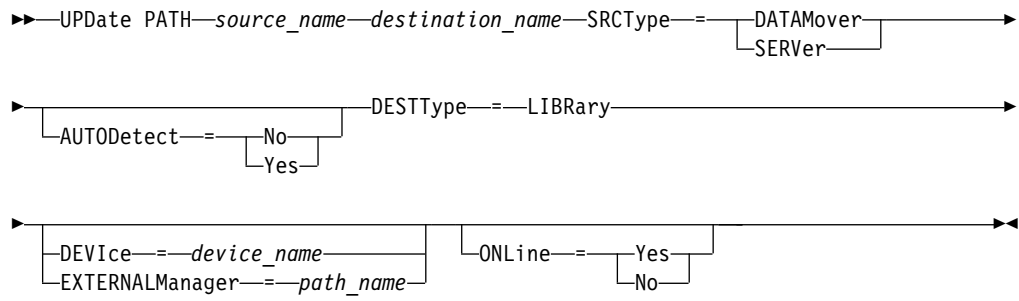
## UPDATE PATH (Change a path when the destination is a library)

Use this syntax when updating a path definition to a library.

### Privilege class

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

### Syntax



### 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.

**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 is automatically detected, reported, and updated in Tivoli Storage Manager. This parameter is optional. This parameter is only valid for paths defined from the local server to a library. Possible values are:

**No** Specifies that the serial number is not automatically updated.

##### **Yes**

Specifies that the serial number is automatically updated to reflect the same serial number that the drive reports to 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. Depending on the capabilities of the device, the AUTODETECT parameter may not be supported.

**DESTType=LIBRARY (Required)**

Specifies that a library is the destination.. This parameter is required.

**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 432 for examples.

*Table 432. Examples of device names*

| Source to destination       | Example         |
|-----------------------------|-----------------|
| Server to a library         | /dev/rmt/tsm1b4 |
| NAS data mover to a library | mc0             |

**Important:**

- For information about the device name when the source is a storage agent, see the Tivoli Storage Manager for Storage Area Networks product information.
- For 349X libraries, the alias name is a symbolic name that is specified in the /etc/ibmatl.conf file. For more information, see *IBM Tape Device Drivers Installation and User's Guide*, which can be downloaded from the IBM Systems support site at <http://www.ibm.com/support/docview.wss?uid=ssg1S7002972>.
- 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 the device name for a library:

```
sysconfig -m
```

**EXTERNALManager**

Specifies the location of the external library manager where Tivoli Storage Manager can send media access requests. Use single quotation marks around the value of this parameter. For example, enter:

```
/opt/GESedt-acsls/bin/elmdt
```

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.

**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.



## 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 433. 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.                                                       |

*Table 433. Commands related to UPDATE POLICYSET (continued)*

| <b>Command</b>     | <b>Description</b>                                                                                   |
|--------------------|------------------------------------------------------------------------------------------------------|
| 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 434. 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.     |

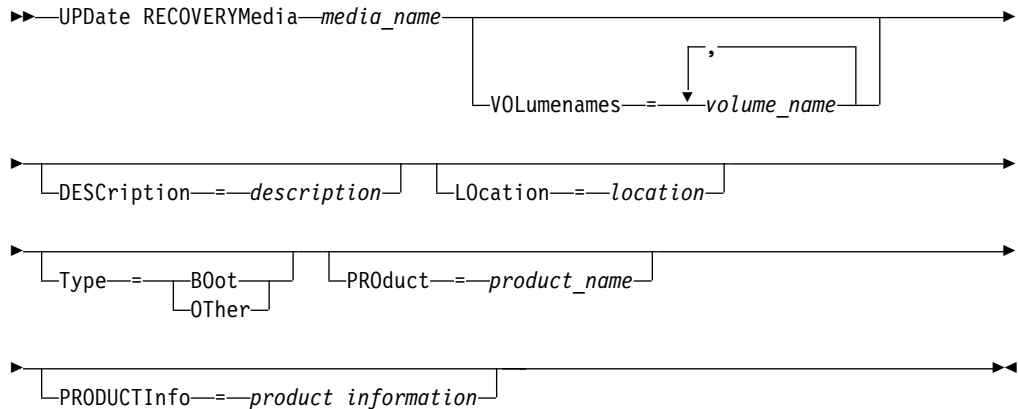
## UPDATE RECOVERYMEDIA (Update recovery media)

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 435. 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 REPLRULE (Update replication rules)

Use this command to enable or disable a replication rule.

Issue this command on the server that acts as a source for replicated data.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—UPDate REPLRule—rule_name—State—=—┌—ENabled—┐—————►►  
└—DISabled—┘
```

### Parameters

#### *rule\_name* (Required)

Specifies the name of the replication rule to be updated. You can use wildcard characters to specify one or more rules. You can specify one of the following rules:

- ALL\_DATA
- ACTIVE\_DATA
- ALL\_DATA\_HIGH\_PRIORITY
- ACTIVE\_DATA\_HIGH\_PRIORITY

#### State (Required)

Specifies whether replication is allowed for the rule. You can specify one of the following values:

##### ENabled

Specifies that the data to which the rule applies is ready to be replicated

##### DISabled

Specifies that replication does not occur until you enable it.

### Example: Disable replication for backup data

Disable replication of normal-priority, active-backup data for all file spaces in all client nodes that are configured for replication:

```
update replrule active_data state=disabled
```

### Related commands

Table 436. Commands related to UPDATE REPLRULE

| Command           | Description                                                             |
|-------------------|-------------------------------------------------------------------------|
| 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 REPLICATION | Displays information about node replication processes.                  |
| QUERY REPLRULE    | Displays information about node replication rules.                      |

Table 436. Commands related to UPDATE REPLRULE (continued)

| <b>Command</b>        | <b>Description</b>                                                 |
|-----------------------|--------------------------------------------------------------------|
| SET ARREPLRULEDEFAULT | Specifies the server node-replication rule for archive data.       |
| SET BKREPLRULEDEFAULT | Specifies the server node-replication rule for backup data.        |
| SET SPREPLRULEDEFAULT | Specifies the server node-replication rule for space-managed data. |
| UPDATE FILESPACE      | Changes file-space node-replication rules.                         |
| UPDATE NODE           | Changes the attributes that are associated with a client node.     |
| VALIDATE REPLICATION  | Verifies replication for file spaces and data types.               |

## 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.

- “UPDATE SCHEDULE (Update an administrative schedule)” on page 1368
- “UPDATE SCHEDULE (Update a client schedule)” on page 1355

*Table 437. Commands related to UPDATE SCHEDULE*

| <b>Command</b>        | <b>Description</b>                                                                              |
|-----------------------|-------------------------------------------------------------------------------------------------|
| 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 MAXSCHEDESESSIONS | 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.

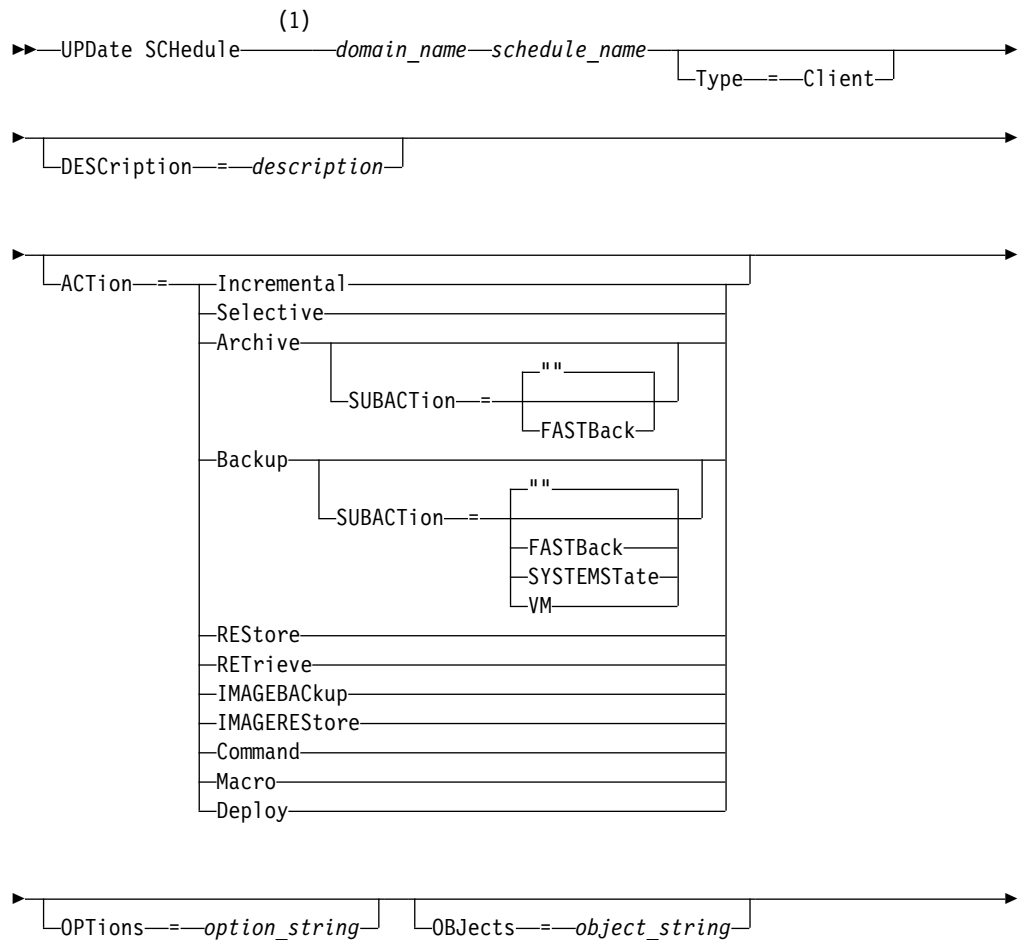
This command does not change the client associations that have been made to this schedule. Any clients that are associated with the original schedule, process the modified schedule.

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

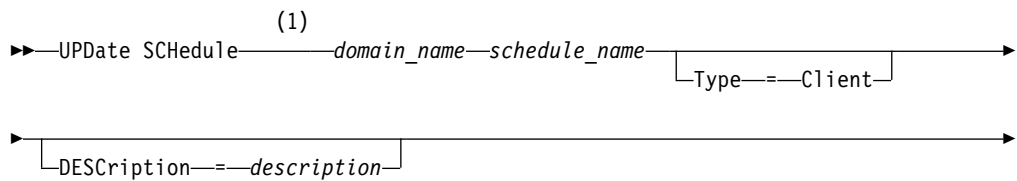


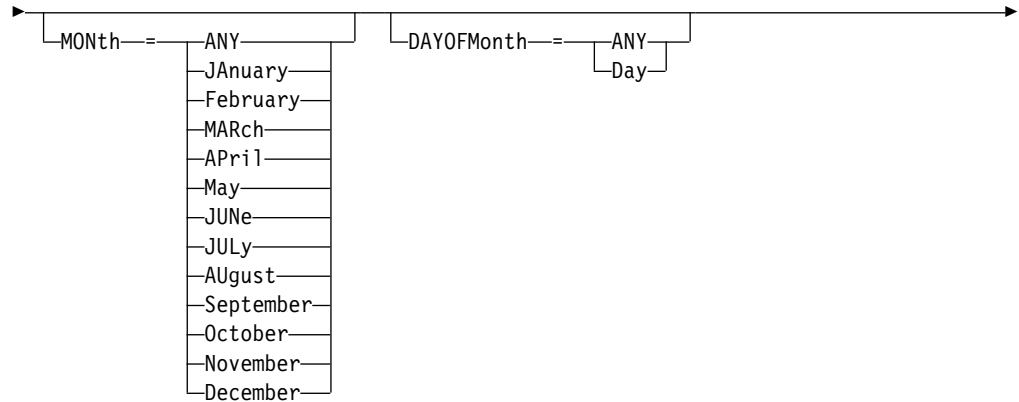
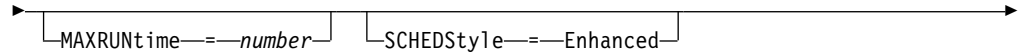
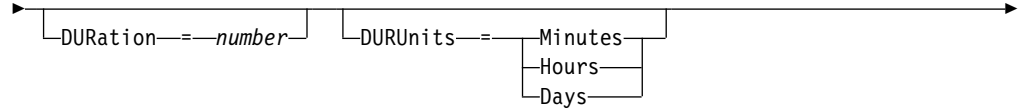
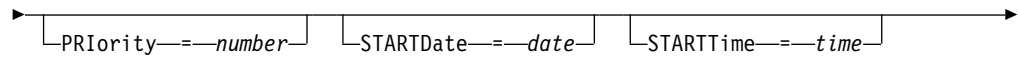
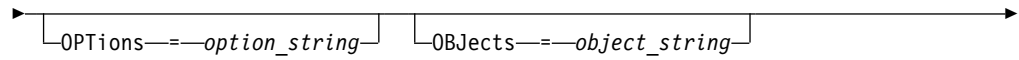
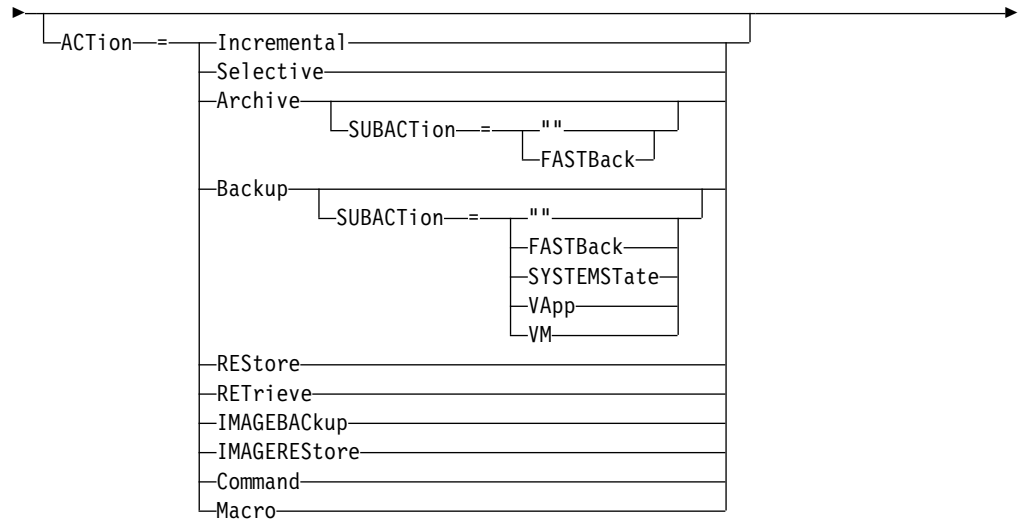


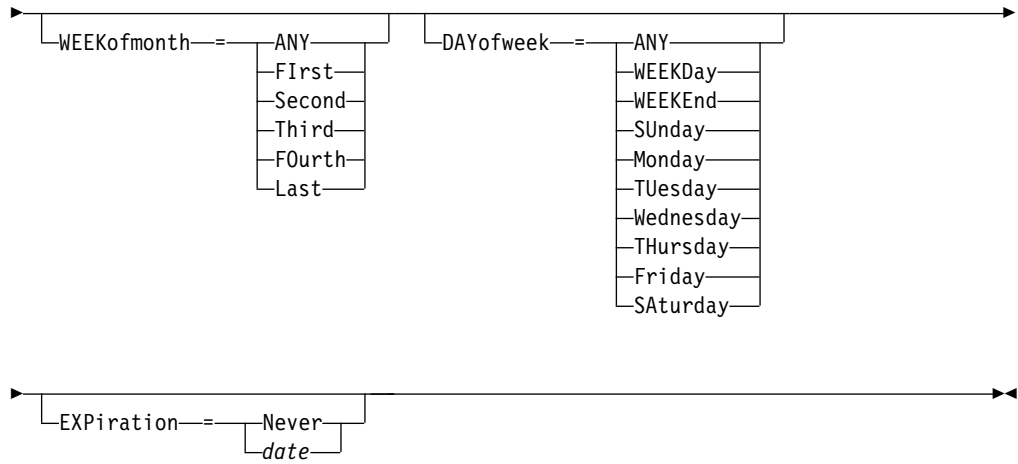
**Notes:**

- 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.

**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 might 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.

**Backup**

Specifies that the schedule backs up 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**

Indicates that the schedule retrieves files that are specified with the **OBJECTS** parameter.

**Remember:** A second file that is specified acts as the retrieve destination. If you need to retrieve multiple groups of files, create a separate schedule for each group of files.

**IMAGEBACKup**

Specifies that the schedule backs up logical volumes that are specified with the **OBJECTS** parameter.

**IMAGERESore**

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.

**SUBACTio**

You can specify one of the following values:

**""** When a null string (two double quotes) is specified with **ACTION=BACKUP** the backup is an incremental.

**FASTBac**

Specifies that a FastBack client operation that is identified by the **ACTION** parameter is to be scheduled for processing. The **ACTION** parameter must be either **ARCHIVE** or **BACKUP**.

**SYSTEMSt**

Specifies that a client Systemstate backup is scheduled.

**VApp**

Specifies that a client vApp backup is scheduled. A vApp is a collection of pre-deployed virtual machines.

**VM** Specifies that a client VMware backup operation is scheduled.

**Deplo**

Specifies whether to update client workstations with deployment packages that are specified with the **OBJECTS** parameter. The **OBJECTS** parameter must contain two specifications, the package files to retrieve and the location from which to retrieve them. Ensure that the objects are in the order *files location*. For example:

```
define schedule standard deploy_1 action=DEPLOY objects=
"\\IBM_ANR_WIN\c$\tsm\maintenance\client\v6r2\Windows\X32\v620\v6200\*
..\IBM_ANR_WIN\"
```

Values for the following options are restricted when you specify ACTION=DEPLOY:

#### **PERUNITS**

Specify PERUNITS=ONETIME. If you specify PERUNITS=PERIOD, the parameter is ignored.

#### **DURUNITS**

Specify MINUTES, HOURS, or DAYS for the **DURUNITS** parameter. Do not specify **INDEFINITE**.

#### **SCHEDSTYLE**

Specify the default style, CLASSIC.

The **SCHEDULE** command fails if the parameters do not conform to the required parameter values, such as the V.R.M.F.

### **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 can 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 result in an error or are ignored when running the schedule from the server. For example, do not include the following options because they have no impact 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 apostrophes. Enclose individual options that contain spaces in quotation marks. 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. 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 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 /opt/file 2, /opt/gif files, and /opt/my test file, enter:  
OBJECTS='"/opt/file 2" "/opt/gif files" "/opt/my test file"'
- To specify /opt/test file, enter:  
OBJECTS='"/opt/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      |

| Value                          | Description                                                                               | Example                                                                                          |
|--------------------------------|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| TODAY+days or +days            | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3.                                                                                  |
| EOLM (End Of Last Month)       | The last day of the previous month.                                                       | EOLM                                                                                             |
| EOLM-days                      | The last day of the previous month minus days specified.                                  | EOLM-1<br><br>To include files that were active a day before the last day of the previous month. |
| BOTM (Beginning Of This Month) | The first day of the current month.                                                       | BOTM                                                                                             |
| BOTM+days                      | The first day of the current month, plus days specified.                                  | BOTM+9<br><br>To include files that were active on the 10th day of the current month.            |

#### 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. You can specify one of the following values:

### **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.

### **MAXRUNTime**

Specifies the maximum run time, which is the number of minutes during which all client sessions that are started by the scheduled operation should be completed. If sessions are still running after the maximum run time, the server issues a warning message, but the sessions continue to run.

**Tip:** The maximum run time is calculated from the beginning of the startup window and not from the time that sessions start within the startup window.

### **Restrictions:**

- The value of the parameter is not distributed to servers that are managed by an enterprise configuration manager.
- The value of the parameter is not exported by the **EXPORT** command.

The parameter is optional. You can specify a number in the range 0-1440. A value of 0 means that the maximum run time is indefinite, and no warning message is issued. The maximum run time must be greater than the startup window duration, which is defined by the **DURATION** and **DURUNITS** parameters.

For example, if the start time of a scheduled operation is 9:00 PM, and the duration of the startup window is 2 hours, the startup window is 9:00 PM - 11:00 PM. If the maximum run time is 240 minutes, that is, 4 hours, all client

sessions for this operation should be completed by 1:00 AM. If one or more sessions are still running after 1:00 AM, the server issues a warning message.

**Tip:** Alternatively, you can specify a *run time alert* value of 1:00 AM in the Tivoli Storage Manager Operations Center.

### **SCHEDStyle**

This parameter is optional. SCHEDSTYLE defines either the interval between times when a schedule can run, or the days on which it can 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 are reset. DAYOFWEEK, PERIOD, and PERUNITS are 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 are reset. MONTH, DAYOFMONTH, WEEKOFMONTH, and DAYOFWEEK are 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. You can specify one of the following values:

#### **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 was 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, which means that the schedule runs 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. You can specify multiple values separated by commas and no intervening blanks. If you specify multiple values, the schedule runs on each of the specified days of the month. If multiple values resolve to the same day, the schedule runs only once that day.

The default value is ANY, which means that the schedule runs 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 runs during each of the specified weeks of the month. If multiple values resolve to the same week, the schedule runs only once during that week.

The default value is ANY. ANY means that the schedule runs 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. You can specify one of the following values:

**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 the priority of a schedule**

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 the expiration date of a schedule**

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 are reset to ANY.

## 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.

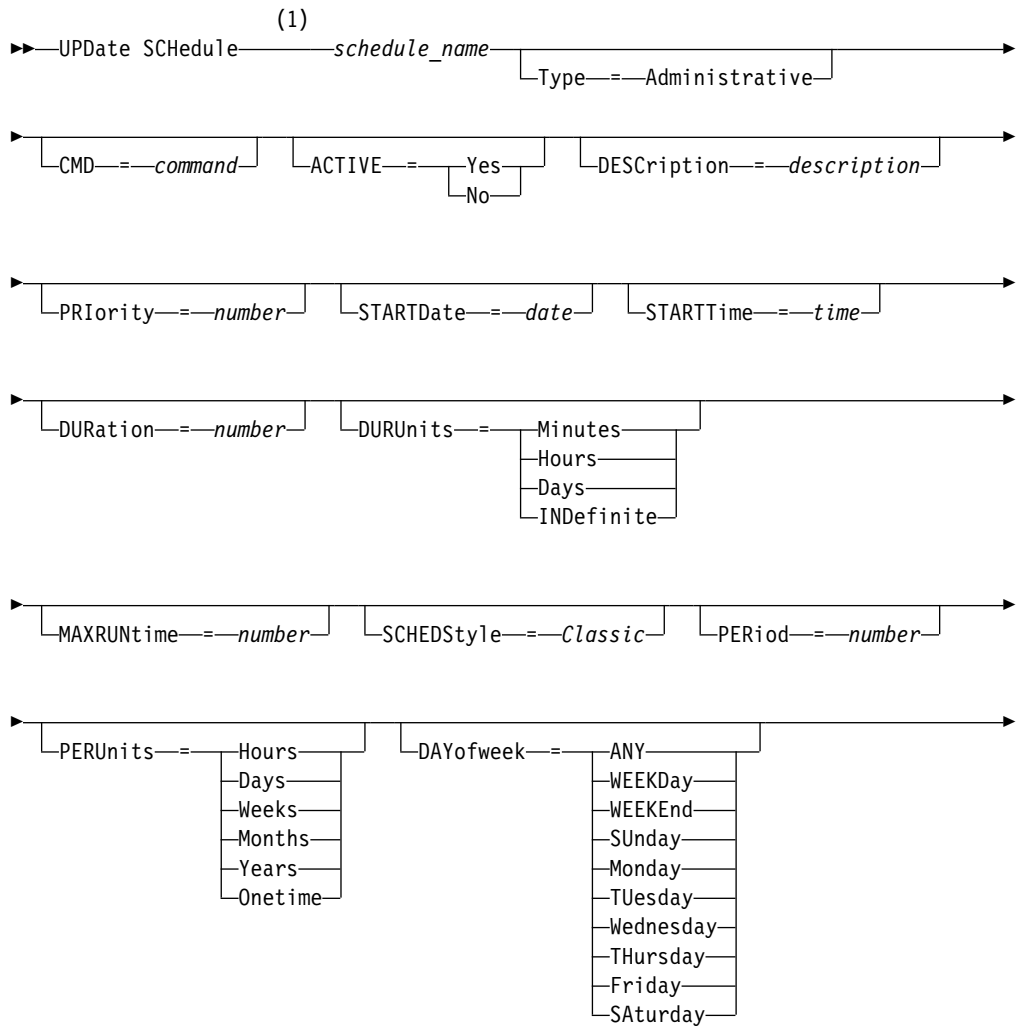
A managed administrative schedule that is updated by a configuration manager is set to an inactive state on the managed servers during configuration refresh processing. It remains in an inactive state until it is updated to an active state on those servers.

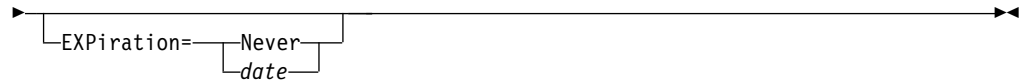
### Privilege class

To update an administrative schedule, you must have system privilege.

### Syntax

#### Classic administrative schedule



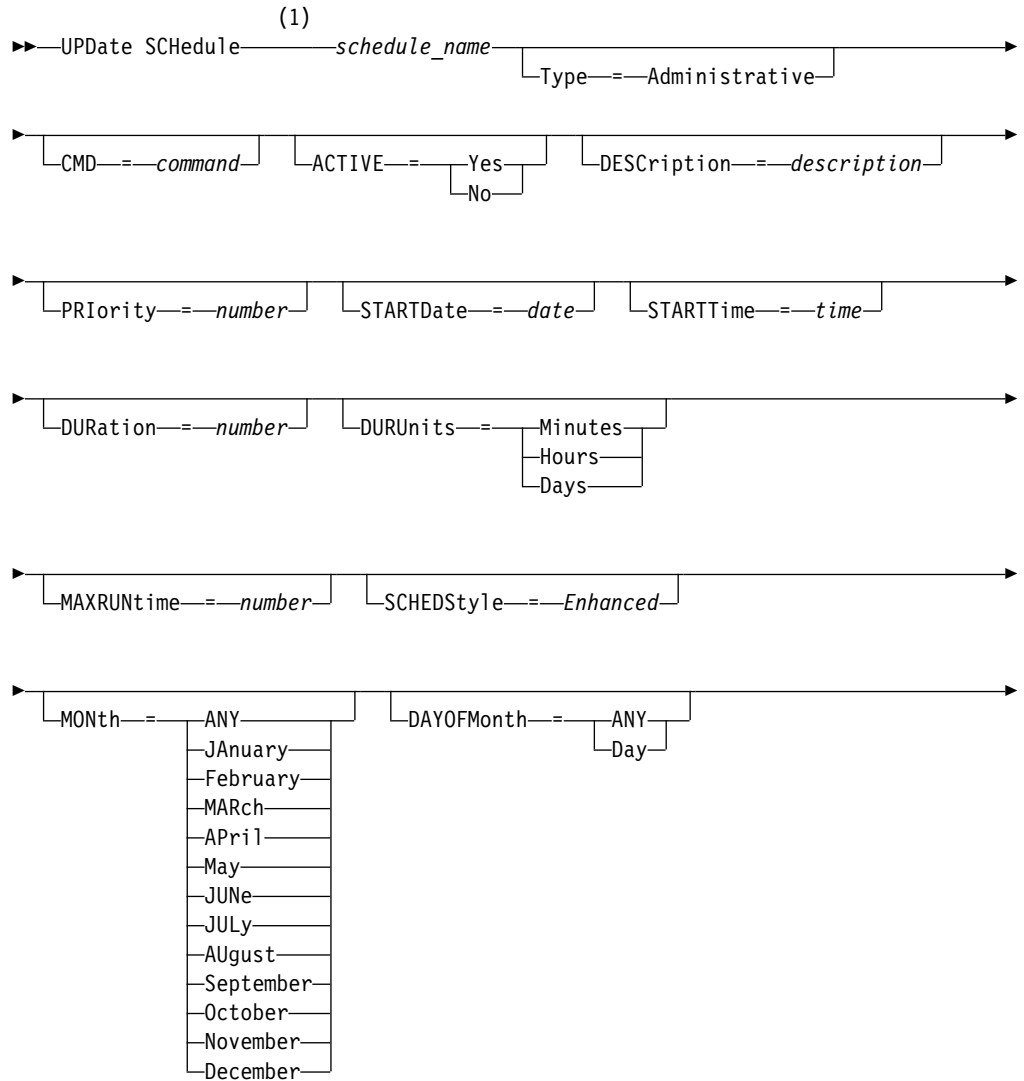


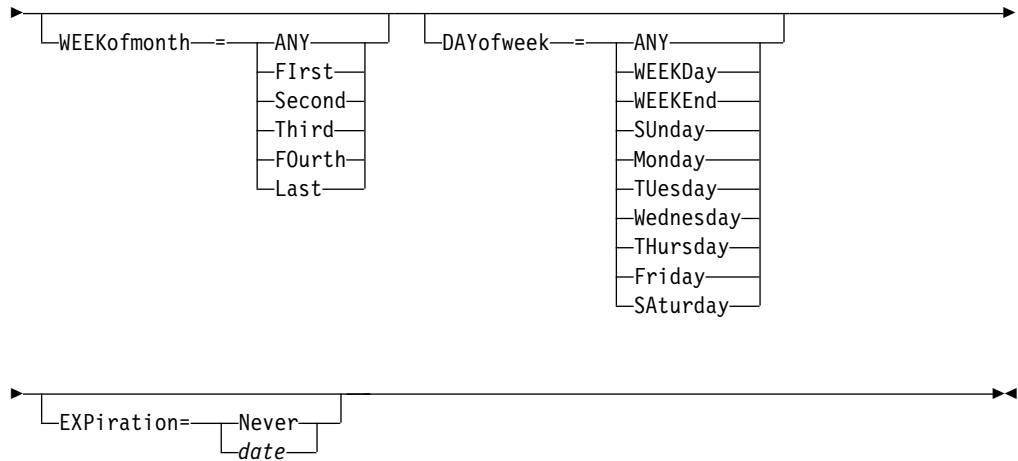
**Notes:**

- 1 You must specify at least one optional parameter on this command.

**Syntax**

**Enhanced administrative schedule**





**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<br><i>+days</i> | The current date plus days specified. The maximum number of days you can specify is 9999. | TODAY +3 or +3.                                                                                  |
| EOLM (End Of Last Month)             | The last day of the previous month.                                                       | EOLM                                                                                             |
| <i>EOLM-days</i>                     | The last day of the previous month minus days specified.                                  | EOLM-1<br><br>To include files that were active a day before the last day of the previous month. |
| BOTM (Beginning Of This Month)       | The first day of the current month.                                                       | BOTM                                                                                             |
| <i>BOTM+days</i>                     | The first day of the current month, plus days specified.                                  | BOTM+9<br><br>To include files that were active on the 10th day of the current month.            |

### 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><br>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><br>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**.

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. You can specify one of the following values:

**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.

**MAXRUNTime**

Specifies the maximum run time, which is the number of minutes during which server processes that are started by the scheduled commands must be completed. If processes are still running after the maximum run time, the central scheduler cancels the processes.

**Tips:**

- The processes might not end immediately when the central scheduler cancels them; they end when they register the cancellation notification from the central scheduler.
- The maximum run time is calculated beginning from when the server process starts. If the schedule command starts more than one process, each process maximum run time is calculated from when the process starts.

- This parameter does not apply to some processes, such as duplicate-identification processes, which can continue to run after the maximum run time.
- This parameter does not apply if the scheduled command does not start a server process.
- Another cancel time might be associated with some commands. For example, the **MIGRATE STGPOOL** command can include a parameter that specifies the length of time that the storage pool migration runs before the migration is automatically canceled. If you schedule a command for which a cancel time is defined, and you also define a maximum run time for the schedule, the processes are canceled at whichever cancel time is reached first.

#### **Restrictions:**

- The value of the parameter is not distributed to servers that are managed by an enterprise configuration manager.
- The value of the parameter is not exported by the **EXPORT** command.

This parameter is optional. You can specify a number in the range 0-1440. A value of 0 means that the maximum run time is indefinite, and the central scheduler does not cancel processes. The maximum run time must be greater than the startup window duration, which is defined by the **DURATION** and **DURUNITS** parameters.

For example, if the start time of a scheduled command is 9:00 PM, and the duration of the startup window is 2 hours, the startup window is 9:00 PM - 11:00 PM. If the maximum run time is 240 minutes, that is, 4 hours, all applicable server processes that are started by the command must be completed by 1:00 AM. If one or more applicable processes are still running after 1:00 AM, the central scheduler cancels the processes.

**Tip:** Alternatively, you can specify an *end time* of 1:00 AM in the Tivoli Storage Manager Operations Center.

#### **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. You can specify one of the following values:

#### **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 was 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. You can specify one of the following values:

**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 SCRATCHPADENTRY (Update a scratch pad entry)

Use this command to update data on a line in the scratch pad.

### Privilege class

To issue this command, you must have system privilege.

### Syntax

```
►►—UPDate SCRATCHPadentry—major_category—minor_category—subject—►►
►Line—==—number—Data—==—data—►►
```

### Parameters

#### *major\_category* (Required)

Specifies the major category in which data is to be updated. This parameter is case sensitive.

#### *minor\_category* (Required)

Specifies the minor category in which data is to be updated. This parameter is case sensitive.

#### *subject* (Required)

Specifies the subject under which data is to be updated. This parameter is case sensitive.

#### **Line (Required)**

Specifies the number of the line on which data is to be updated.

#### **Data (Required)**

Specifies the new data to be stored on the line. Previous data is deleted. You can enter up to 1000 characters. Enclose the data in quotation marks if the data contains one or more blanks. The data is case sensitive.

### Example: Update a scratch pad entry

Update the vacation contact details of an administrator, Jane, in a database that stores information about the location of all administrators:

```
update scratchpadentry admin_info location jane line=2 data=
"Out of the office until 18 Nov."
```

### Related commands

Table 438. Commands related to UPDATE SCRATCHPADENTRY

| Command                 | Description                                                              |
|-------------------------|--------------------------------------------------------------------------|
| DEFINE SCRATCHPADENTRY  | Creates a line of data in the scratch pad.                               |
| DELETE SCRATCHPADENTRY  | Deletes a line of data from the scratch pad.                             |
| QUERY SCRATCHPADENTRY   | Displays information that is contained in the scratch pad.               |
| SET SCRATCHPADRETENTION | Specifies the amount of time for which scratch pad entries are retained. |



## 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.

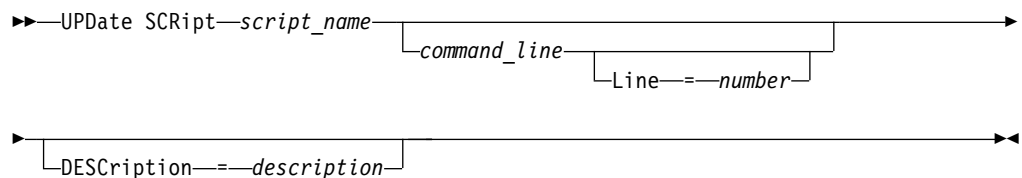
**Restriction:** You cannot redirect the output of a command within a Tivoli Storage Manager script. Instead, run the script and then specify command redirection. For example, to direct the output of **script1** to the `c:\temp\test.out` directory, run the script and specify command redirection as in the following example:

```
run script1 > c:\temp\test.out
```

### 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.

You have the options of running commands serially, in parallel, or serially and in parallel by specifying the **SERIAL** or **PARALLEL** script commands for this parameter. You can run multiple commands in parallel and wait for them to complete before proceeding to the next command. Commands will run serially until the parallel command is encountered.

Conditional logic flow statements can be used. These statements include IF, EXIT, and GOTO.

#### **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 qsample "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 qsample "query stgpool" 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 qsample "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 439. Commands related to **UPDATE SCRIPT**

| Command       | Description                                            |
|---------------|--------------------------------------------------------|
| COPY SCRIPT   | Creates a copy of a script.                            |
| DEFINE SCRIPT | Defines a script to the Tivoli Storage Manager server. |

*Table 439. Commands related to UPDATE SCRIPT (continued)*

| <b>Command</b> | <b>Description</b>                                      |
|----------------|---------------------------------------------------------|
| 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.

**Restriction:** If this server 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 that is stored on the corresponding target server. Changing the server name by using the **SET SERVERNAME** command might have additional implications, varying by operating system. The following are some examples:

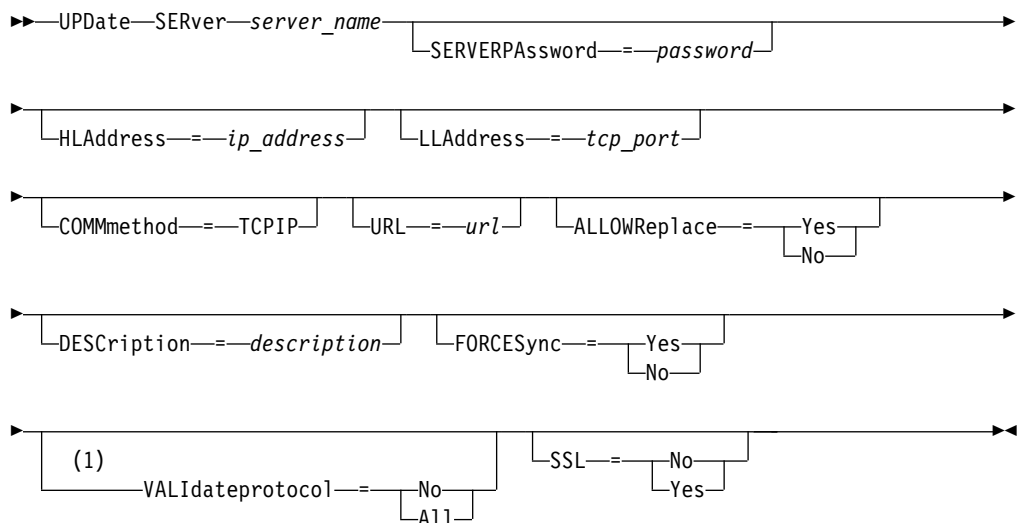
- Passwords might be invalidated
- Device information might be affected
- Registry information about Windows operating systems might change

### Privilege class

To issue this command, you must have system privilege.

### Syntax for:

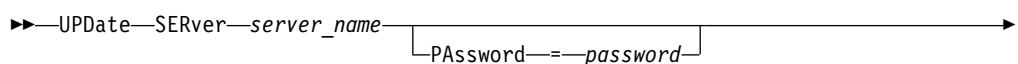
- Enterprise configuration
- Enterprise event logging
- Command routing
- Storage agent
- Node replication source and target servers

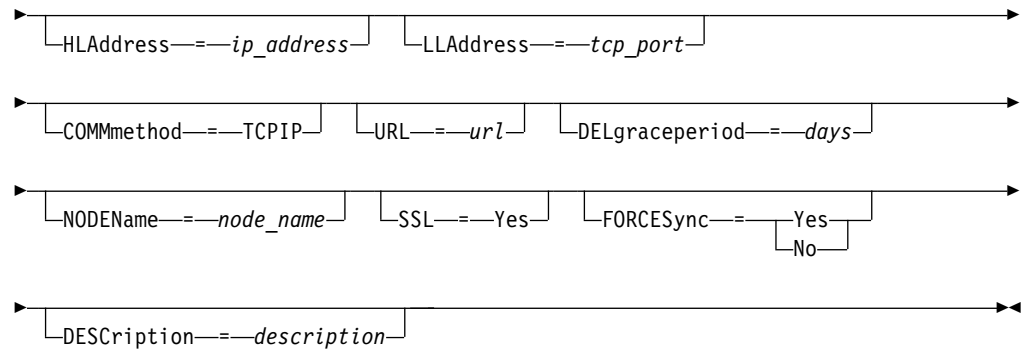


### Notes:

- 1 The VALIDATEPROTOCOL parameter applies only 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 that is 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 the address in the TCPPOINT server option of the target server. When SSL=YES, the port must already be designated for SSL communications on the target server.

### COMMmethod

Specifies the communication method that is used to connect to the server. This parameter is optional.

### URL

Specifies the URL address that is used to access this server from the Administration Center. The parameter is optional.

### DELgraceperiod

Specifies a number of days that an object remains on the target server after it was marked for deletion. You can specify a value 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.

### 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 enables a source server to 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. You can specify one of the following values:

**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 a cyclic redundancy check to validate the data sent between the storage agent and Tivoli Storage Manager server. The parameter is optional. The default is NO. You can specify one of the following values:

**No** Specifies that data validation is not to be performed on any data sent between the storage agent and server.

**All**

Specifies that data validation is to 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 affects performance as additional resources are required to calculate and compare CRC values between the storage agent and the server.

**ALLOWReplac**

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. You can specify one of the following values:

**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.

**SSL**

Specifies the communication mode of the server. You can specify one of the following values:

**No** Specifies that unencrypted communication with the specified server occurs over a TCP/IP session without SSL.

**Yes**

Specifies an SSL session for communication with the specified server. The following conditions apply if you specify YES:

- Before starting the servers, self-signed certificates of the partner servers must be in the key database file (cert.kdb) of each of the servers.
- You can define multiple server names with different parameters for the same target server.
- SSL support is active if the server options file contains the SSLTCPPOINT or SSLTCPADMINPOINT option or if a server is defined with SSL=YES at startup.

## Example: Update a deletion grace period for a server

Update the definition of SERVER2 to specify that objects remain on the target server for 10 days after they were marked for deletion.

```
update server server2 delgraceperiod=10
```

## Example: Update the URL for a server

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 440. 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.                                                                                                                                                                          |
| DELETE FILESPACE  | Deletes data associated with client file spaces. If a file space is part of a collocation group and you remove the file space from a node, the file space is removed from the collocation group. |
| 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 node 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 that are 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 server 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 441. 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──┘
                               |
                               └──STGPPOOL—==—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 STGPPOOL 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.

## 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 442. 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 STATUSTHRESHOLD (Update a status monitoring threshold)

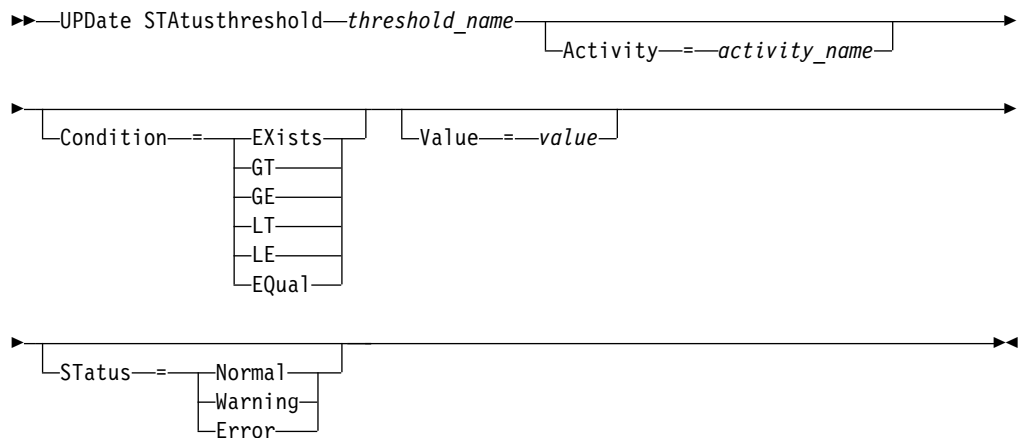
Use this command to update an existing status monitoring threshold.

Status monitoring thresholds compare the defined conditions to the status monitoring server queries and inserts the results in the status monitoring table.

Multiple thresholds can be defined for an activity. For example, you can create a threshold that provides a warning status if storage pool capacity utilization is greater than 80%. You can then create another threshold that provides error status if storage pool capacity utilization is greater than 90%.

**Note:** If a threshold is already defined for an EXISTS condition, you cannot define another threshold with one of the other condition types.

### Syntax



### Parameters

#### *threshold\_name* (Required)

Specifies the threshold name that you want to update. The name cannot exceed 48 characters in length.

#### *activity* (Required)

Specify this value to change the activity for an existing threshold. Specify one of the following values:

##### **PROCESSSUMMARY**

Specifies the number of processes that are currently active.

##### **SESSIONSUMMARY**

Specifies the number of sessions that are currently active.

##### **CLIENTSESSIONSUMMARY**

Specifies the number of client sessions that are currently active.

##### **SCHEDCLIENTSESSIONSUMMARY**

Specifies the number of scheduled client sessions.

##### **DBUTIL**

Specifies the database utilization percentage. The default warning threshold value is 80%, and the default error threshold value is 90%.

**DBFREESPACE**

Specifies the free space available in the database in gigabytes.

**DBUSEDSPACE**

Specifies the amount of database space that is used, in gigabytes.

**ARCHIVELOGFREESPACE**

Specifies the free space that is available in the archive log, in gigabytes.

**STGPOOLUTIL**

Specifies the storage pool utilization percentage. The default warning threshold value is 80%, and the default error threshold value is 90%.

**STGPOOLCAPACITY**

Specifies the storage pool capacity in gigabytes.

**AVGSTGPOOLUTIL**

Specifies the average storage pool utilization percentage across all storage pools. The default warning threshold value is 80%, and the default error threshold value is 90%.

**TOTSTGPOOLCAPACITY**

Specifies the total storage pool capacity in gigabytes for all available storage pools.

**TOTSTGPOLS**

Specifies the number of defined storage pools.

**TOTRWSTGPOLS**

Specifies the number of defined storage pools that are readable or writeable.

**TOTNOTRWSTGPOLS**

Specifies the number of defined storage pools that are not readable or writeable.

**STGPOOLINUSEANDDEFINED**

Specifies the total number of defined volumes that are in use.

**ACTIVELOGUTIL**

Specifies the current percent utilization of the active log. The default warning threshold value is 80%, and the default error threshold value is 90%.

**ARCHLOGUTIL**

Specifies the current utilization of the archive log. The default warning threshold value is 80%, and the default error threshold value is 90%.

**CPYSTGPOOLUTIL**

Specifies the percent utilization for a copy storage pool. The default warning threshold value is 80%, and the default error threshold value is 90%.

**PMRYSTGPOOLUTIL**

Specifies the percent utilization for a primary storage pool. The default warning threshold value is 80%, and the default error threshold value is 90%.

**DEVCLASSPCTDRVOFFLINE**

Specifies the percent utilization of drives that are offline, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTDRVPOLLING**

Specifies the drives polling, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTLIBPATHSOFFLINE**

Specifies the library paths that are offline, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTPATHSOFFLINE**

Specifies the percentage of device class paths that are offline, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTDISKSNOTRW**

Specifies the percentage of disks that are not writable for the disk device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**DEVCLASSPCTDISKSUNAVAILABLE**

Specifies the percentage of the disk volumes that are unavailable, by device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**FILEDEVCLASSPCTSCRUNALLOCATABLE**

Specifies the percentage of scratch volumes that the server cannot allocate for a given non-shared file device class. The default warning threshold value is 25%, and the default error threshold value is 50%.

**Condition**

Specify this value to change the condition of an existing threshold. This parameter is optional. Specify one of the following values:

**EXists**

Creates a status monitoring indicator if the activity exists.

**GT** Creates a status monitoring indicator if the activity outcome is greater than the specified value.

**GE** Creates a status monitoring indicator if the activity outcome is greater than or equal to the specified value.

**LT** Creates a status monitoring indicator if the activity outcome is less than the specified value.

**LE** Creates a status monitoring indicator if the activity outcome is less than or equal to the specified value.

**EQual**

Creates a status monitoring indicator if the activity outcome is equal to the specified value.

**Value**

Specify this parameter to change the value that is compared with the activity output for the specified condition. You can specify an integer in the range 0 - 999999999999999.

**Status**

Specify this value to change the status of the indicator that is created in status monitoring if the condition that is being evaluated passes. This parameter is optional. Specify one of the following values:

**Normal**

Specifies that the status indicator has a normal status value.

**Warning**

Specifies that the status indicator has a warning status value.

**Error**

Specifies that the status indicator has an error status value.

## Update an existing status threshold

Update a status threshold for average storage pool utility percentage by issuing the following command:

```
update statusthreshold avgstgpl "AVGSTGPOOLUTIL" value=90 condition=gt status=error
```

## Related commands

Table 443. Commands related to **UPDATE STATUSTHRESHOLD**

| Command                                                                                                              | Description                                                                 |
|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| "DELETE STATUSTHRESHOLD (Delete a status monitoring threshold)" on page 442                                          | Deletes a status monitoring threshold.                                      |
| "QUERY MONITORSTATUS (Query the monitoring status)" on page 807                                                      | Displays information about monitoring alerts and server status settings.    |
| "QUERY MONITORSETTINGS (Query the configuration settings for monitoring alerts and server status)" on page 804       | Displays information about monitoring alerts and server status settings.    |
| "QUERY STATUSTHRESHOLD (Query status monitoring thresholds)" on page 932                                             | Displays information about a status monitoring thresholds.                  |
| "SET STATUSMONITOR (Specifies whether to enable status monitoring)" on page 1178                                     | Specifies whether to enable status monitoring.                              |
| "SET STATUSATRISKINTERVAL (Specifies whether to enable client at-risk activity interval evaluation)" on page 1176    | Specifies whether to enable client at-risk activity interval evaluation     |
| "SET STATUSREFRESHINTERVAL (Set refresh interval for status monitoring)" on page 1180                                | Specifies the refresh interval for status monitoring.                       |
| "SET STATUSSKIPASFAILURE (Specifies whether to use client at-risk skipped files as failure evaluation)" on page 1182 | Specifies whether to use client at-risk skipped files as failure evaluation |
| "UPDATE STATUSTHRESHOLD (Update a status monitoring threshold)" on page 1389                                         | Changes the attributes of an existing status monitoring threshold.          |

## UPDATE STGPOOL (Update a storage pool)

Use this command to change a storage pool.

**Restriction:** If a client is using the simultaneous-write function and data deduplication, the data deduplication feature is disabled during backups to a storage pool.

The UPDATE STGPOOL command takes six forms. The syntax and parameters for each form are defined separately.

- “UPDATE STGPOOL (Update a primary random access storage pool)” on page 1395
- “UPDATE STGPOOL (Update a primary sequential access pool)” on page 1405
- “UPDATE STGPOOL (Update a copy sequential access storage pool)” on page 1420
- “UPDATE STGPOOL (Update an active-data sequential access)” on page 1427

*Table 444. 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 COLLOGROUP         | Defines a collocation group.                                                                  |
| DEFINE COLLOCMEMBER       | Adds a client node or file space 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 or file space from a collocation group.                                 |
| DELETE STGPOOL            | Deletes a storage pool from server storage.                                                   |
| MOVE DRMEDIA              | Moves DRM media onsite and offsite.                                                           |
| MOVE MEDIA                | Moves storage pool volumes that are managed by an automated library.                          |
| QUERY COLLOGROUP          | 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.                                     |

*Table 444. Commands related to UPDATE STGPOOL (continued)*

| <b>Command</b>    | <b>Description</b>                                     |
|-------------------|--------------------------------------------------------|
| SHRED DATA        | Manually starts the process of shredding deleted data. |
| UPDATE COLLOGROUP | Updates the description of a collocation group.        |



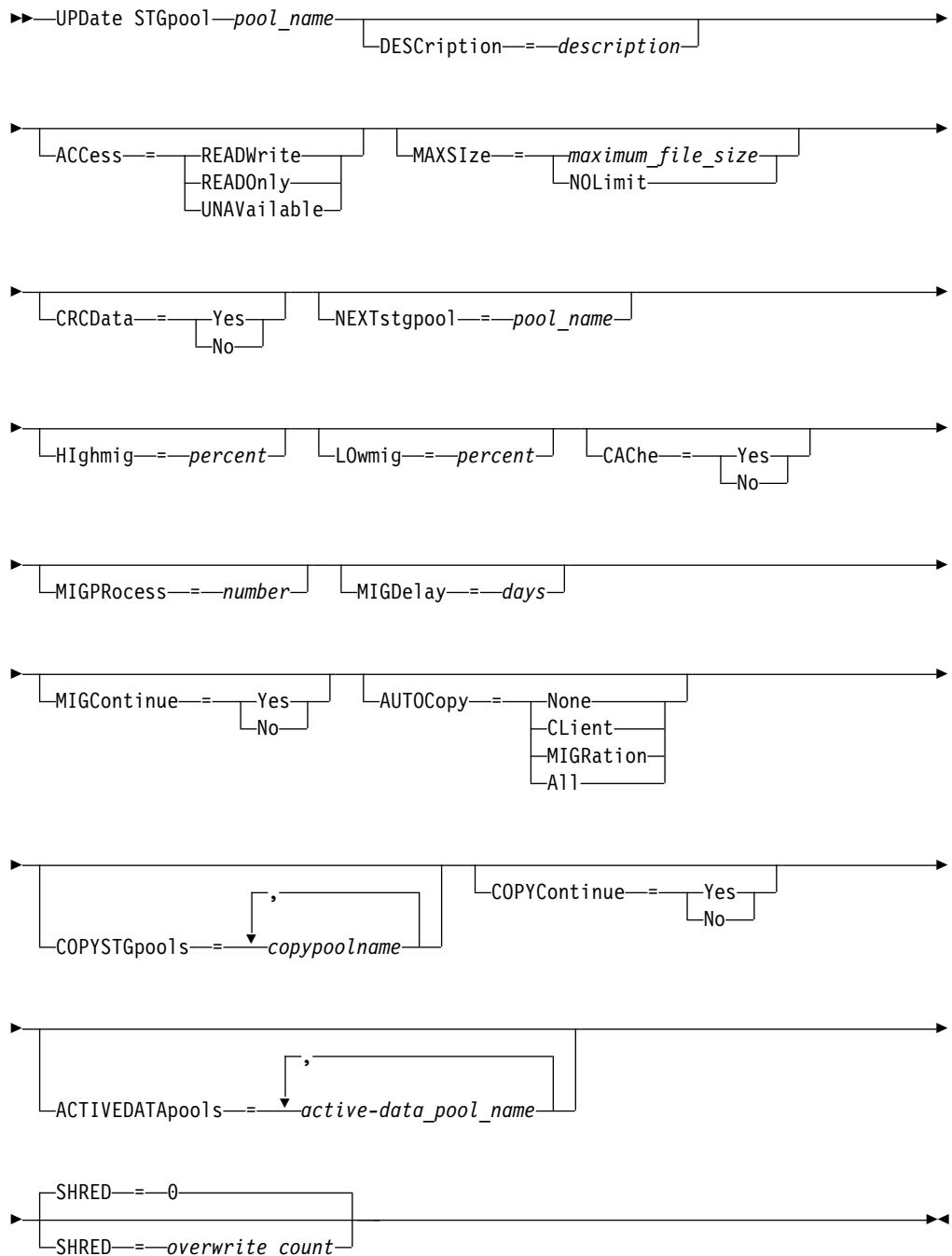
## UPDATE STGPOOL (Update a primary random access storage pool)

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. You can specify the following values:

#### 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 was 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 was 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. You can specify the following values:

#### 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 terabytes, 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

| Scale factor | Meaning  |
|--------------|----------|
| G            | gigabyte |
| T            | terabyte |

The client estimates the size of files that are sent to the server. The client estimate is used rather than the actual amount of data that is sent to the server. Client options, such as deduplication, compression, and encryption, can cause the actual amount of data that is sent to the server to be larger or smaller than the size estimate. For example, the compression of a file might be smaller in size than the estimate, thus sending less data than the estimate. Furthermore, a binary file might be larger in size after the compression processing, thus sending more data than the estimate.

See the following table for information about where a file is stored when its size exceeds the **MAXSIZE** parameter.

Table 445. Where a file is stored according to the file size and the pool that is specified

| File size                | Pool specified                                                 | Result                                                                            |
|--------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------|
| Exceeds the maximum size | No pool is specified as the next storage pool in the hierarchy | The server does not store the file                                                |
|                          | A pool is specified as the next storage pool in the hierarchy  | The server stores the file in the next storage pool that can accept the file size |

If you specify the next storage pool parameter, define one storage pool in your hierarchy to have no limit on the maximum file size. 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 multiple files that are sent in a single transaction, the server considers the size of the transaction to be the file size. If the total size of all files in the transaction is larger than the maximum size limit, the server does not store the files in the storage pool.

#### **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 that is stored in your storage hierarchy. You can specify the following values:

##### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more expenditure 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 you do not specify a next storage pool, the following actions occur:

- The server cannot migrate files from this storage pool
- The server cannot store files that exceed the maximum size for this storage pool in another storage pool

**Restrictions:**

- To ensure that you do not create a chain of storage pools that leads to an endless loop, specify at least one storage pool in the hierarchy with no value.
- If you specify a sequential-access pool as the next storage pool, the pool must be in either NATIVE or NONBLOCK data format.
- Do not specify a directory-container or cloud-container storage pool.
- Do not use this parameter to specify a storage pool for data migration.

**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 0 - 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. You can specify an integer 0 - 99 for this optional parameter.

When migration is by node or file space, depending upon collocation, the level of the storage pool can fall below the value that you specified for this parameter. To empty the storage pool, set **LOWMIG=0**.

**CACHE**

Specifies whether the migration process leaves a cached copy of a file in this storage pool after you migrate the file to the next storage pool. This parameter is optional. You can specify the following values:

**Yes**

Specifies that caching is enabled.

**No**

Specifies that caching is disabled.

Using cache might improve your ability to retrieve files, but might affect the performance of other processes.

**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 1 - 999.

During migration, these processes are run in parallel to provide the potential for improved migration rates.

**Tips:**

- The number of migration processes is dependent upon the following settings:
  - The setting of the **MIGPROCESS** parameter
  - The collocation setting of the next pool

- The number of nodes or the number of collocation groups with data in the storage pool that is being migrated

For this example, `MIGPROCESS =6`, the next pool `COLLOCATE` parameter is `NODE`, but there are only two nodes with data on the storage pool. Migration processing consists of only two processes, not six. If the `COLLOCATE` parameter is `GROUP` group and both nodes are in the same group, migration processing consists of only one process. If the `COLLOCATE` parameter is `NO` or `FILESPEC` group, and each node has two file spaces with backup data, then migration processing consists of only four processes.

- When you specify this parameter, consider whether the simultaneous-write function is enabled for server data migration. Each migration process requires a mount point and a drive for each copy storage pool and active-data pool that is defined to the target storage pool.

### **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 following items:

- The number of days that the file was in the storage pool
- The number of days, if any, since the file was retrieved by a client

The lesser of the two values are compared to the specified `MIGDELAY` value. For example, if all the following conditions are true, a file is not migrated:

- A file was in a storage pool for five days.
- The file was accessed by a client within the past three days.
- The value that is specified for the `MIGDELAY` parameter is four days.

This parameter is optional. You can specify an integer 0 - 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 that are based 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.

You can specify one of the following values:

#### **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.

#### **AUTOCopy**

Specifies when Tivoli Storage Manager runs simultaneous-write operations to copy storage pools and active-data pools. This parameter affects the following operations:

- Client store sessions
- Server import processes
- Server data-migration processes

If an error occurs while data is being simultaneously written to a copy storage pool or active-data pool during a migration process, the server stops writing to the failing storage pools for the remainder of the process. However, the server continues to store files into the primary storage pool and any remaining copy storage pools or active-data pools. These pools remain active for the duration of the migration process. Copy storage pools are specified using the **COPYSTGPOLLS** parameter. Active-data pools are specified using the **ACTIVEDATAPOLLS** parameter.

You can specify one of the following values:

#### **None**

Specifies that the simultaneous-write function is disabled.

#### **CLient**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions or server import processes. During server import processes, data is written simultaneously to only copy storage pools. Data is not written to active-data pools during server import processes.

#### **MIGRation**

Specifies that data is written simultaneously to copy storage pools and active-data pools only during migration to this storage pool. During server data-migration processes, data is written simultaneously to copy storage pools and active-data pools only if the data does not exist in those pools. Nodes whose data is being migrated must be in a domain associated with an active-data pool. If the nodes are not in a domain associated with an active pool, the data cannot be written to the pool.

#### **All**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions, server import processes, or server data-migration processes. Specifying this value ensures that data is written simultaneously whenever this pool is a target for any of the eligible operations.

#### **COPYSTGpools**

Specifies the names of copy storage pools where the server simultaneously writes data. You can specify a maximum of three copy pool names that are separated by commas. Spaces between the names of the copy pools are not allowed. 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 **COPYSTGPOLLS=COPY1,COPY2,COPY3**. To remove all existing copy storage pools that are associated with the primary storage pool, specify a null string (""), for the value (for example, **COPYSTGPOLLS=""**).

When you specify a value for the **COPYSTGPPOOLS** parameter, you can also specify a value for the **COPYCONTINUE** parameter. For more information, see the **COPYCONTINUE** parameter.

The combined total number of storage pools that are specified in the **COPYSGTPOOLS** 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 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.

The server can write data simultaneously to copy storage pools for the following operations:

- Back up and archive operations by Tivoli Storage Manager backup-archive clients or application clients that are 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:** The simultaneous-write function is not supported for the following store operations:

- When the operation is using LAN-free data movement. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is accepted.
- 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 that are defined:
  - The copy storage pools are ignored
  - The data is stored into the primary storage pool only

**Attention:** The function that is provided by the **COPYSTGPPOOLS** parameter is not intended to replace the **BACKUP STGPOOL** command. If you use the **COPYSTGPPOOLS** 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 reacts to a copy storage pool write failure for any of the copy storage pools that are listed in the **COPYSTGPPOOLS** parameter. This parameter is optional. When you specify the **COPYCONTINUE** parameter, either a **COPYSTGPPOOLS** list must exist or the **COPYSTGPPOOLS** parameter must also be specified.

You can specify the following values:

#### **Yes**

If the **COPYCONTINUE** parameter 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.

**No** If the **COPYCONTINUE** parameter is set to **NO**, the server will fail the current transaction and discontinue the store operation.

**Restrictions:**

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of the active-data pools, the server stops writing to the failing active-data pool for the remainder of the session, but continues storing files into the primary pool and any remaining active-data pools and copy storage pools. The active-data pool list is active only for the life of the session and applies to all the primary storage pools in a particular storage pool hierarchy.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server import. If data is being written simultaneously and a write failure occurs to the primary storage pool or any copy storage pool, the server import process fails.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server data migration. If data is being written simultaneously and a write failure occurs to any copy storage pool or active-data pool, the failing storage pool is removed and the data migration process continues. Write failures to the primary storage pool cause the migration process to fail.

**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 allowed.

The combined total number of storage pools that are specified in the **COPYSGTPools** 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 that is 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.

The server can write data simultaneously to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients that use 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. Writing data simultaneously to active-data pools is not supported when you use LAN-free data movement. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is followed.
3. The simultaneous-write function 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 that are defined:



- The active-data pools are ignored
  - The data is stored into the primary storage pool only
4. You cannot use the simultaneous-write function with CENTERA storage devices.
  5. Data that is being imported is not stored in active-data pools. After an import operation, use the **COPY ACTIVE DATA** command to store the imported data in an active-data pool.

**Attention:** The function that is provided by the **ACTIVEDATAPOOLES** parameter is not intended to replace the **COPY ACTIVE DATA** command. If you use the **ACTIVEDATAPOOLES** parameter, use the **COPY ACTIVE DATA** command to ensure that the active-data pools contain all active data of the primary storage pool.

## **SHRED**

Specifies whether data is physically overwritten when it is deleted. This parameter is optional. You can specify an integer 0 - 10.

If you specify a value of zero, the server deletes the data from the database. However, the storage that is used to contain the data is not overwritten, and the data exists in storage until that storage is reused for other data. It might be possible to discover and reconstruct the data after it is deleted. Changing the value (for example, resetting it to 0) does not affect data that was deleted and is waiting to be overwritten.

If you specify a value greater than 0, the server deletes the data both logically and physically. The server overwrites the storage that is used to contain the data the specified number of times. This overwriting increases the difficulty of discovering and reconstructing the data after it is deleted.

To ensure that all copies of the data are shredded, specify a **SHRED** value greater than zero for the storage pool that is specified in the **NEXTSTGPOOL** parameter. Do not specify either the **COPYSTGPOOLS** or **ACTIVEDATAPOOLES**. Specifying relatively high values for the overwrite count generally improves the level of security, but might affect performance adversely.

Overwriting of deleted data is done asynchronously after the delete operation is complete. Therefore, the space that is occupied by the deleted data remains occupied for some time. The space is not 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. Existing cached files 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 are not shredded when they are erased.

**Important:** After an export operation finishes and identifies files for export, any change 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 canceling 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 allow caching**

Update the random access storage pool that is named BACKUPPOOL to allow 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.

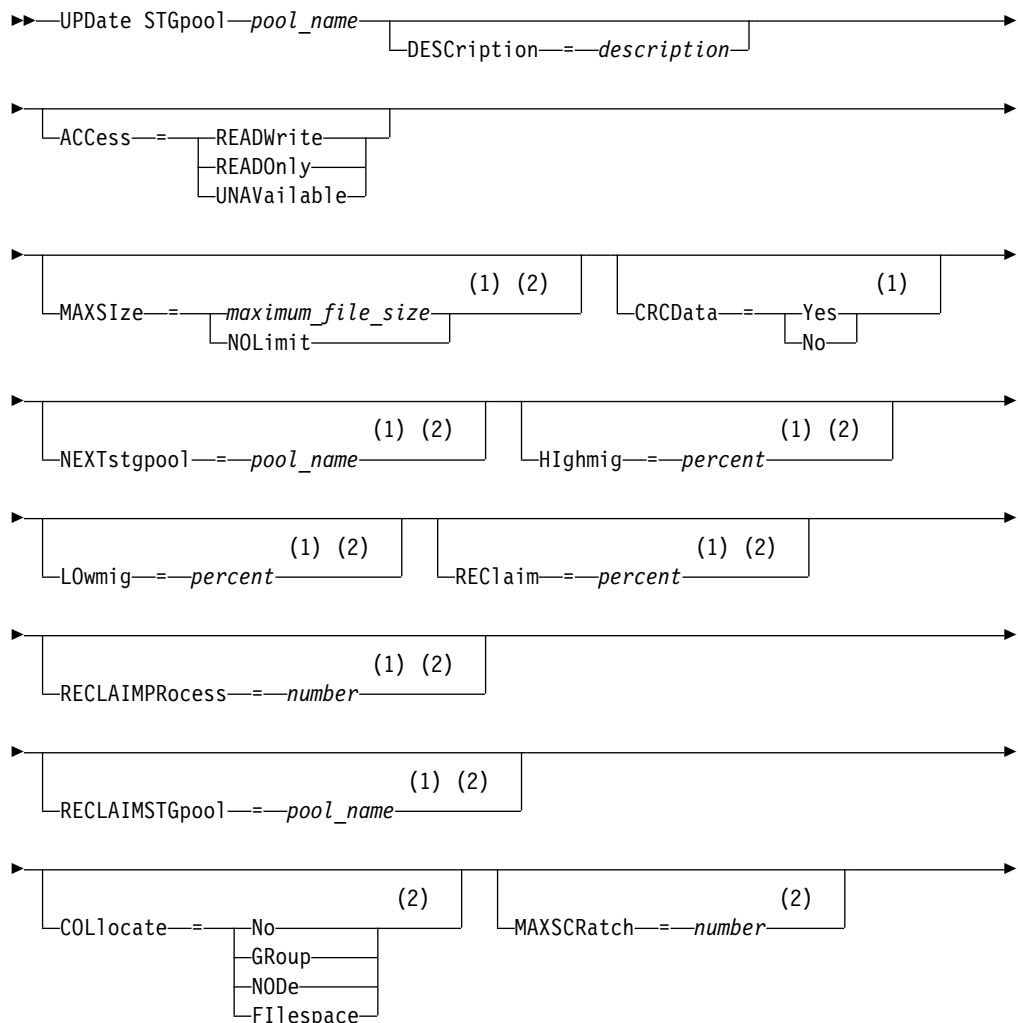
### Restrictions:

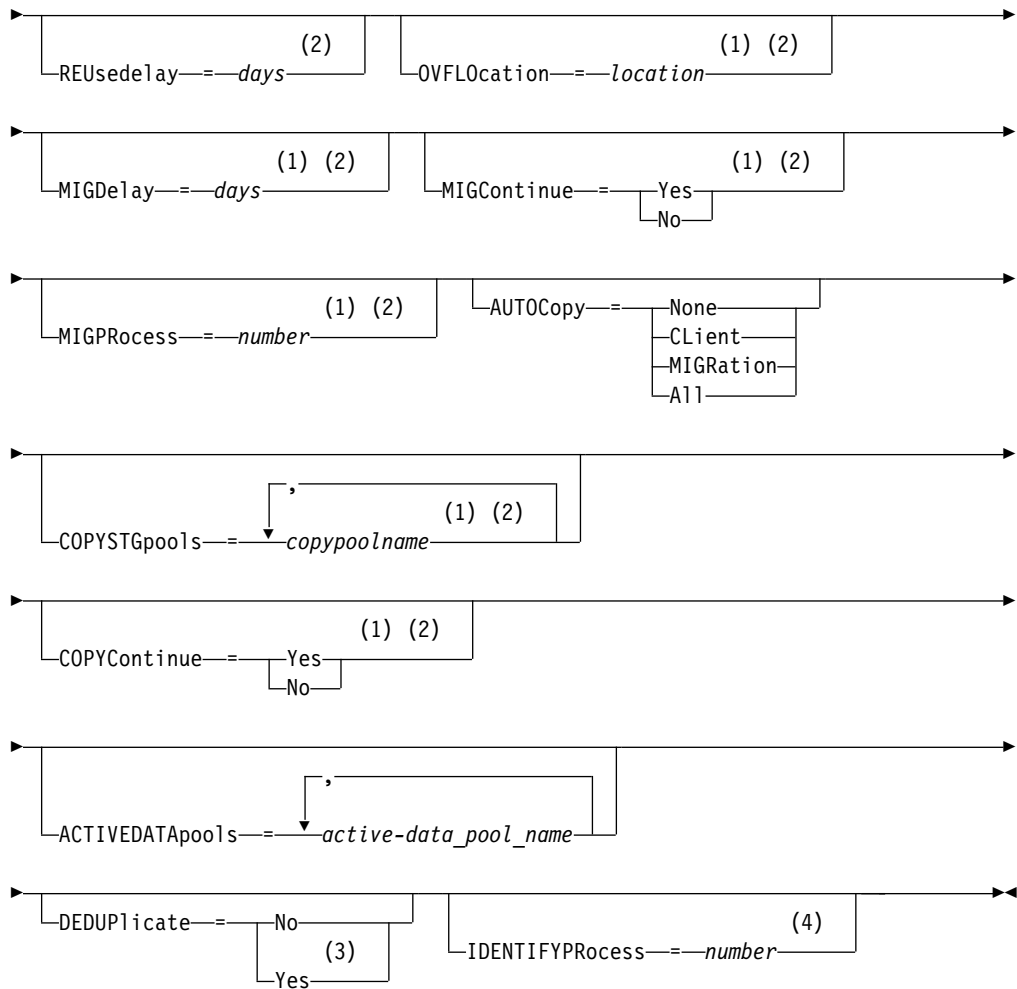
1. You cannot use this command to change the data format for the storage pool.
2. If the value for DATAFORMAT is NETAPPDUMP, CELERRADUMP, or NDMPDUMP, you can modify only the following attributes:
  - DESCRIPTION
  - ACCESS
  - COLLOCATE
  - MAXSCRATCH
  - REUSEDELAY

### 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. You can specify the following values:

### **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 was 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 was 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. You can specify the following values:

### **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 terabytes, 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

The client estimates the size of files that are sent to the server. The client estimate is used rather than the actual amount of data that is sent to the server. Client options, such as deduplication, compression, and encryption, can cause the actual amount of data that is sent to the server to be larger or smaller than the size estimate. For example, the compression of a file might be smaller in

size than the estimate, thus sending less data than the estimate. Furthermore, a binary file might be larger in size after the compression processing, thus sending more data than the estimate.

When the physical size of the storage pool exceeds the **MAXSIZE** parameter, the following table shows where files are typically stored.

*Table 446. The location of a file according to the file size and the pool that is specified*

| File size                | Pool specified                                                 | Result                                                                            |
|--------------------------|----------------------------------------------------------------|-----------------------------------------------------------------------------------|
| Exceeds the maximum size | No pool is specified as the next storage pool in the hierarchy | The server does not store the file                                                |
|                          | A pool is specified as the next storage pool in the hierarchy  | The server stores the file in the next storage pool that can accept the file size |

**Tip:** If you also specify the **NEXTstgpool** parameter, define one storage pool in your hierarchy to have no limit on the maximum file size by specifying the **MAXSize=NOLimit** parameter. When you have at least one pool with no size limit, you ensure that no matter what its size, the server can store the file.

For multiple files that are sent in a single transaction, the server considers the size of the transaction to be the file size. If the total size of all files in the transaction is larger than the maximum size limit, the server does not store the files in the storage pool.

**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 that is stored in your storage hierarchy. You can specify the following values:

##### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more processing is required to calculate and compare CRC values between the storage pool and the server.

**No** Specifies that data is stored without CRC information.

**Tip:** For storage pools that are associated with the 3592, LTO, or ECARTRIDGE device type, logical block protection provides better protection against data corruption than CRC validation for a storage pool. If you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

To enable logical block protection, specify a value of READWRITE for the **LBPROTECT** parameter on the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands

for the 3592, LTO, or ECARTRIDGE device types. Logical block protection is supported only on the following types of drives and media:

- IBM LTO5 and later.
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.
- Oracle StorageTek T10000C and T10000D drives.

### **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 completes 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 might fail because, at the time the actual storage operation is attempted, the space is no longer available.

### **Restrictions:**

- To ensure that you do not create a chain of storage pools that leads to an endless loop, specify at least one storage pool in the hierarchy with no value.
- If you specify a sequential-access pool as the next storage pool, the pool must be in either NATIVE or NONBLOCK data format.
- Do not specify a directory-container or cloud-container storage pool.
- Do not use this parameter to specify a storage pool for data migration.
- 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 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 0 - 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 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 0 - 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 allow 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, which is based on the percentage of reclaimable space on a volume. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the Tivoli Storage Manager database.

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 1 - 100.

The server determines that the volume is a candidate for reclamation if the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool.

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 1 - 999. You can specify one or more reclamation processes for each primary sequential-access storage pool.

When you calculate the value for this parameter, consider the following resources that are required for reclamation processing:

- The number of sequential storage pools.
- The number of logical and physical drives that can be dedicated to the operation.

To access sequential volumes, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive.

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 **RECLAIMSTGPPOOL** parameter is not specified or that the reclaim storage pool has the same device class as the storage pool that is 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.

**Restriction:** This parameter is not available for storage pools that use the following data formats:

- NETAPPDUMP
- CELERRADUMP
- NDMPDUMP

#### **RECLAIMSTGpool1**

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 that are 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, which is stored on as few volumes as possible, that belong to one of the following candidates:

- A single client node

- A group of file spaces
- A group of client nodes
- A client file space

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 that is needed to collocate files for storing and the number of volumes required. Collocation can also impact the number of processes that are migrating disks to sequential pool.

You can specify one of the following options:

**No** Specifies that collocation is disabled. During migration from disk, processes are created at a file space level.

**GRoup**

Specifies that collocation is enabled at the group level for client nodes or file spaces. For collocation groups, the server attempts to put data for nodes or file spaces that belong to the same collocation group on as few volumes as possible.

If you specify COLLOCATE=GROUP but do not define any collocation groups, or if you do not add nodes or file spaces to a collocation group, data is collocated by node. Consider tape usage when you organize client nodes or file spaces into collocation groups.

For example, if a tape-based storage pool consists of data from nodes and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates the data by group for grouped nodes. Whenever possible, the server collocates data that belongs 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 that are associated with a group.
- Collocates the data by node 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.
- During migration from disk, the server creates migration processes at the collocation group level for grouped nodes, and at the node level for ungrouped nodes.

If a tape-based storage pool consists of data from grouped file spaces and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates by group, the data for grouped file spaces only. Whenever possible, the server collocates data that belongs to a group of file spaces on a single tape or on as few tapes as possible. Data for a single file space can also be spread across several tapes that are associated with a group.
- Collocates the data by node (for file spaces that are not explicitly defined to a file space collocation group). For example, node1 has file spaces that are named A, B, C, D, and E. File spaces A and B belong to a file space collocation group but C, D, and E do not. File spaces A and B are collocated by file space collocation group, while C, D, and E are collocated by node.

- During migration from disk, the server creates migration processes at the collocation group level for grouped file spaces.

Data is collocated on the least number of sequential access volumes.

#### **NODe**

Specifies that collocation is enabled at the client node level. For collocation groups, 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 try to collocate those file spaces. For compatibility with an earlier version, 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 is collocated by node.

For COLLOCATE=NODE, the server creates processes at the node level when you migrate data from disk.

#### **FILEspace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to place data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to place data for different file spaces on different volumes.

For COLLOCATE=FILESPEC, the server creates processes at the file space level when you migrate data from disk.

#### **MAXSCRatch**

Specifies the maximum number of scratch volumes that the server can request. This parameter is optional. You can specify an integer 0 - 100000000. By allowing the server to request scratch volumes, you avoid having to define each volume to be used.

The value that is 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 use 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 is not reached. The 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 0 - 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 can 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 0 - 9999.

If you want the server to count the number of days that are 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.

You can specify one of the following values:

##### **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 1 - 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 161.

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.

**Note:** When you specify this parameter, consider whether the simultaneous-write function is enabled for server data migration. Each migration process requires a mount point and a drive for each copy storage pool and active-data pool that is defined to the target storage pool.

#### **AUTOCopy**

Specifies when Tivoli Storage Manager completes simultaneous-write operations. This parameter affects the following operations:

- Client store sessions
- Server import processes
- Server data-migration processes

If the AUTOCOPY option is set to ALL or CLIENT, and there is at least one storage pool that is listed in the COPYSTGPools or ACTIVE DATAPools options, any client-side deduplication is disabled.

If an error occurs while data is being simultaneously written to a copy storage pool or active-data pool during a migration process, the server stops writing to the failing storage pools for the remainder of the process. However, the server continues to store files into the primary storage pool and any remaining copy storage pools or active-data pools. These pools remain active for the duration of the migration process. Copy storage pools are specified using the **COPYSTGPools** parameter. Active-data pools are specified using the **ACTIVEDATAPools** parameter.

You can specify one of the following values:

**None**

Specifies that the simultaneous-write function is disabled.

**CLient**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions or server import processes. During server import processes, data is written simultaneously to only copy storage pools. Data is not written to active-data pools during server import processes.

**MIGRation**

Specifies that data is written simultaneously to copy storage pools and active-data pools only during migration to this storage pool. During server data-migration processes, data is written simultaneously to copy storage pools and active-data pools only if the data does not exist in those pools. Nodes whose data is being migrated must be in a domain associated with an active-data pool. If the nodes are not in a domain associated with an active pool, the data cannot be written to the pool.

**All**

Specifies that data is written simultaneously to copy storage pools and active-data pools during client store sessions, server import processes, or server data-migration processes. Specifying this value ensures that data is written simultaneously whenever this pool is a target for any of the eligible operations.

**COPYSTGpools**

Specifies the names of copy storage pools where the server simultaneously writes data. You can specify a maximum of three copy pool names that are separated by commas. Spaces between the names of the copy pools are not allowed. 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 that are associated with the primary storage pool, specify a null string ("") for the value (for example, **COPYSTGPools=""**).

When you specify a value for the **COPYSTGPools** parameter, you can also specify a value for the **COPYCONTINUE** parameter. For more information, see the **COPYCONTINUE** parameter.

The combined total number of storage pools that are specified in the **COPYSGTPools** 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 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.

The server can write data simultaneously to copy storage pools during the following operations:

- Back up and archive operations by Tivoli Storage Manager backup-archive clients or application clients that use 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 operations takes precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is accepted.
3. The simultaneous-write function 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 that is provided by the **COPYSTGPOLS** parameter is not intended to replace the **BACKUP STGPOOL** command. If you use the **COPYSTGPOLS** 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 reacts to a copy storage pool write failure for any of the copy storage pools that are listed in the **COPYSTGPOLS** parameter. This parameter is optional. The default is YES. When you specify the **COPYCONTINUE** parameter, either a **COPYSTGPOLS** list must exist or the **COPYSTGPOLS** parameter must also be specified.

The **COPYCONTINUE** parameter has no effect on the simultaneous-write function during migration.

You can specify the following values:

**Yes**

If the **COPYCONTINUE** parameter 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.

**No** If the **COPYCONTINUE** parameter is set to **NO**, the server will fail the current transaction and discontinue the store operation.

**Restrictions:**

- The setting of the **COPYCONTINUE** parameter does not affect active-data pools. If a write failure occurs for any of the active-data pools, the server stops writing to the failing active-data pool for the remainder of the session, but continues storing files into the primary pool and any remaining active-data pools and copy storage pools. The active-data pool list is active only for the life of the session and applies to all the primary storage pools in a particular storage pool hierarchy.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server import. If data is being written simultaneously and a write failure occurs to the primary storage pool or any copy storage pool, the server import process fails.
- The setting of the **COPYCONTINUE** parameter does not affect the simultaneous-write function during server data migration. If data is being written simultaneously and a write failure occurs to any copy storage pool or active-data pool, the failing storage pool is removed and the data migration process continues. Write failures to the primary storage pool cause the migration process to fail.

**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 allowed.

The combined total number of storage pools that are specified in the **COPYSGTPools** 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.

The server can write data simultaneously to active-data pools only during backup operations by Tivoli Storage Manager backup-archive clients or application clients that use 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. Writing data simultaneously to active-data pools is not supported when the operation is using LAN-free data movement. Simultaneous-write operations take precedence over LAN-free data movement, causing the operations to go over the LAN. However, the simultaneous-write configuration is accepted.
3. The simultaneous-write function 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 cannot 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 that is provided by the **ACTIVEDATAPOOLES** parameter is not intended to replace the **COPY ACTIVATEDATA** command. If you use the **ACTIVEDATAPOOLES** 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 is 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 server-side duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a device class associated with the FILE device type. Enter a value 1 - 50.

**Remember:** 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, becomes idle, and then processes five more files, then the total number of files processed is nine. Processes end only when canceled or when the number of duplicate-identification processes for the storage pool is changed to a value less than the number currently specified.

#### **Example: Update the primary sequential storage pool's mountable scratch volumes**

Update the primary sequential storage pool that is named TAPEPOOL1 to allow as many as 10 scratch volumes to be mounted.

```
update stgpool tapepool1 maxscratch=10
```

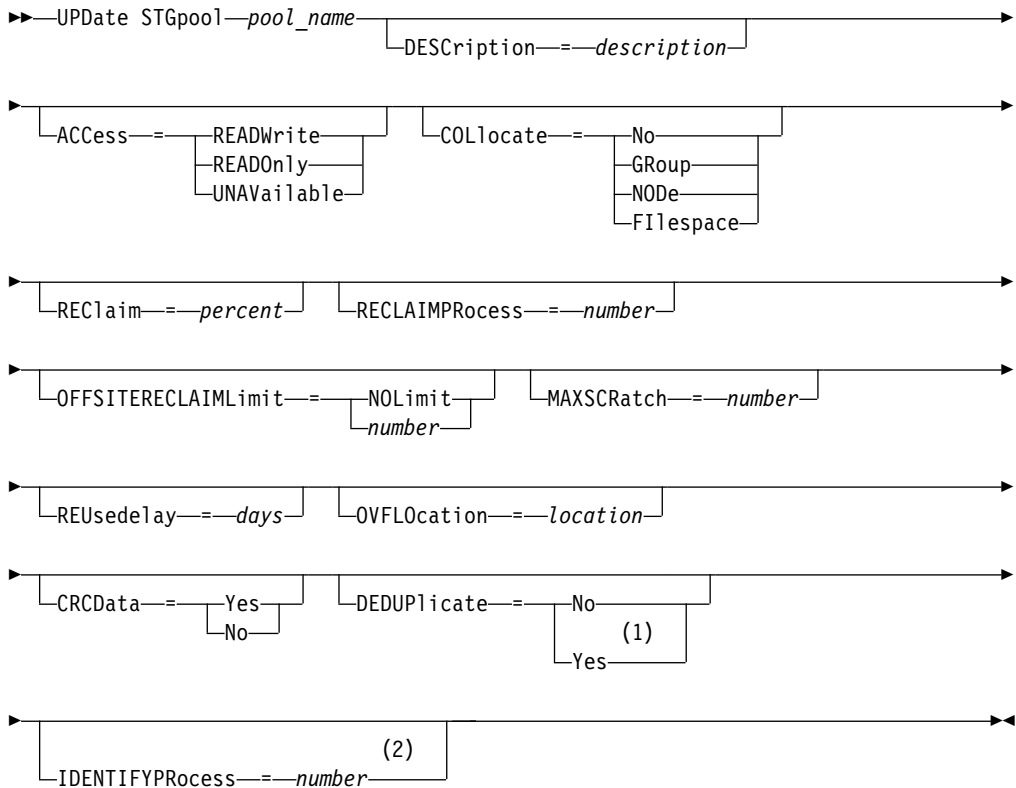
## UPDATE STGPOOL (Update a copy sequential access storage pool)

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. You can specify the following values:

### **READWrite**

Specifies that files can be read from and written to the volumes in the copy storage pool.

### **READOnly**

Specifies that client nodes can read only files that are 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 allowed 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 that are 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 allowed 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, which is stored on as few volumes as possible, that belong to one of the following candidates:

- A single client node
- A group of file spaces
- A group of client nodes
- A client file space

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 that is needed to collocate files for storing and the number of volumes required.

You can specify one of the following options:

**No** Specifies that collocation is disabled.

### **GRoup**

Specifies that collocation is enabled at the group level for client nodes or file spaces. For collocation groups, the server attempts to put data for nodes or file spaces that belong to the same collocation group on as few volumes as possible.

If you specify COLLOCATE=GROUP but do not define any collocation groups, or if you do not add nodes or file spaces to a collocation group, data is collocated by node. Consider tape usage when you organize client nodes or file spaces into collocation groups.

For example, if a tape-based storage pool consists of data from nodes and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates the data by group for grouped nodes. Whenever possible, the server collocates data that belongs 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 that are associated with a group.
- Collocates the data by node 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.

If a tape-based storage pool consists of data from grouped file spaces and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates by group, the data for grouped file spaces only. Whenever possible, the server collocates data that belongs to a group of file spaces on a single tape or on as few tapes as possible. Data for a single file space can also be spread across several tapes that are associated with a group.
- Collocates the data by node (for file spaces that are not explicitly defined to a file space collocation group). For example, node1 has file spaces named A, B, C, D, and E. File spaces A and B belong to a file space collocation group but C, D, and E do not. File spaces A and B are collocated by file space collocation group, while C, D, and E are collocated by node.

Data is collocated on the least amount of sequential access volumes.

#### **NODe**

Specifies that collocation is enabled at the client node level. For collocation groups, 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 try to collocate those file spaces. For compatibility with an earlier version, 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 is collocated by node.

#### **FILEspace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to place data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to place data for different file spaces on different volumes.

#### **RECLaim**

Specifies when the server reclaims a volume, which is based on the percentage of reclaimable space on a volume. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the Tivoli Storage Manager database.

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 1 - 100. The value 100 means that reclamation is not completed.

The server determines that the volume is a candidate for reclamation if the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool.

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 can 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 1 - 999.

When you calculate the value for this parameter, consider the following resources that are required for reclamation processing:

- The number of sequential storage pools.
- The number of logical and physical drives that can be dedicated to the operation.

To access sequential volumes, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive.

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.

You can specify one or more reclamation processes for each copy storage pool. You can specify multiple concurrent reclamation processes for a single copy storage pool, which makes better use of your available tape drives or FILE volumes. If multiple concurrent processing is not necessary, specify a value of 1 for the **RECLAIMPROCESS** parameter.

### **OFFSITERECLAIMLimit**

Specifies the number of offsite volumes that space is reclaimed from during reclamation for this storage pool. This parameter is optional. You can specify the following values:

#### **NOLimit**

Specifies that you want to reclaim the space in all of your offsite volumes.

#### *number*

Specifies the number of offsite volumes to reclaim space from. You can specify an integer 0 - 99999. A value of zero means that none of the offsite volumes are reclaimed.

**Tip:** To determine the value for the **OFFSITERECLAIMLIMIT**, use the statistical information in the message that is issued at the end of the offsite volume reclamation operation. 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 0 - 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

The value that is 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. An administrator can query the server for empty, offsite scratch volumes and return them 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 use 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 is not 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 0 - 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.

**Tip:** Use this parameter to 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 that are specified for this parameter must be the same as the number specified for the **SET DRMDBBACKUPEXPIREDAYS** command.

#### **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 ("").

#### **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 that is stored in your storage hierarchy. You can specify the following values:

##### **Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more processing is required to calculate and compare CRC values between the storage pool and the server.

**No** Specifies that data is stored without CRC information.

**Tip:** For storage pools that are associated with the 3592, LTO, or ECARTRIDGE device type, logical block protection provides better protection against data corruption than CRC validation for a storage pool. If you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

To enable logical block protection, specify a value of READWRITE for the **LBPROTECT** parameter on the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands for the 3592, LTO, or ECARTRIDGE device types. Logical block protection is supported only on the following types of drives and media:

- IBM LTO5 and later.
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.
- Oracle StorageTek T10000C and T10000D drives.

#### **DEDuplicate**

Specifies whether the data that is stored in this storage pool is 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 server-side duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value 1 - 50.

**Remember:** 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, becomes idle, and then processes five more files, then the total number of files processed is nine. Processes end only when canceled or when the number of duplicate-identification processes for the storage pool is changed to a value less than the number 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 that is 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
```



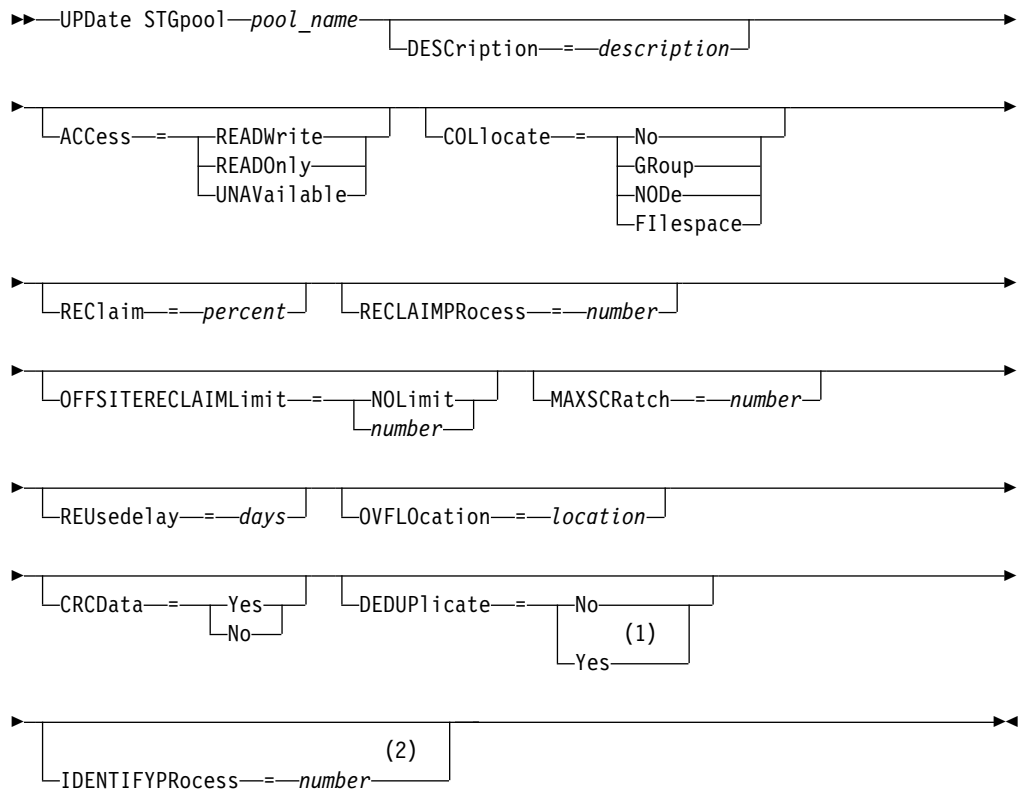
## UPDATE STGPOOL (Update an active-data sequential access)

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. You can specify the following values:

**READWrite**

Specifies that files can be read from and written to the volumes in the active-data pool.

**READOnly**

Specifies that client nodes can read only files that are 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 allowed 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 that are 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 allowed 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, which is stored on as few volumes as possible, that belong to one of the following candidates:

- A single client node
- A group of file spaces
- A group of client nodes
- A client file space

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 that is needed to collocate files for storing and the number of volumes required.

You can specify one of the following options:

**No** Specifies that collocation is disabled.

**GRoup**

Specifies that collocation is enabled at the group level for client nodes or file spaces. For collocation groups, the server attempts to put data for nodes or file spaces that belong to the same collocation group on as few volumes as possible.

If you specify COLLOCATE=GROUP but do not define any collocation groups, or if you do not add nodes or file spaces to a collocation group, data is collocated by node. Consider tape usage when you organize client nodes or file spaces into collocation groups.

For example, if a tape-based storage pool consists of data from nodes and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates the data by group for grouped nodes. Whenever possible, the server collocates data that belongs 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 that are associated with a group.
- Collocates the data by node 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.

If a tape-based storage pool consists of data from grouped file spaces and you specify COLLOCATE=GROUP, the server completes the following actions:

- Collocates by group, the data for grouped file spaces only. Whenever possible, the server collocates data that belongs to a group of file spaces on a single tape or on as few tapes as possible. Data for a single file space can also be spread across several tapes that are associated with a group.
- Collocates the data by node (for file spaces that are not explicitly defined to a file space collocation group). For example, node1 has file spaces named A, B, C, D, and E. File spaces A and B belong to a file space collocation group but C, D, and E do not. File spaces A and B are collocated by file space collocation group, while C, D, and E are collocated by node.

Data is collocated on the least amount of sequential access volumes.

#### **NODe**

Specifies that collocation is enabled at the client node level. For collocation groups, 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 try to collocate those file spaces. For compatibility with an earlier version, 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 is collocated by node.

#### **FIlespace**

Specifies that collocation is enabled at the file space level for client nodes. The server attempts to place data for one node and file space on as few volumes as possible. If a node has multiple file spaces, the server attempts to place data for different file spaces on different volumes.

#### **REClaim**

Specifies when the server reclaims a volume, which is based on the percentage of reclaimable space on a volume. Reclaimable space is the amount of space that is occupied by files that are expired or deleted from the Tivoli Storage Manager database.

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 action makes

the original volume available for reuse. This parameter is optional. You can specify an integer 1 - 100. The value 100 means that reclamation is not completed.

The server determines that the volume is a candidate for reclamation if the percentage of reclaimable space on a volume is greater than the reclamation threshold of the storage pool.

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 can 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 1 - 999.

When you calculate the value for this parameter, consider the following resources that are required for reclamation processing:

- The number of sequential storage pools.
- The number of logical and physical drives that can be dedicated to the operation.

To access sequential volumes, Tivoli Storage Manager uses a mount point and, if the device type is not FILE, a physical drive.

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.

You can specify one or more reclamation processes for each active-data pool. You can specify multiple concurrent reclamation processes for a single active-data pool, which makes better use of your available tape drives or FILE volumes. If multiple concurrent processing is not necessary, specify a value of 1 for the **RECLAIMPROCESS** parameter.

#### **OFFSITERECLAIMLimit**

Specifies the number of offsite volumes that space is reclaimed from during reclamation for this storage pool. This parameter is optional. You can specify the following values:

##### **NOLimit**

Specifies that you want to reclaim the space in all of your offsite volumes.

*number*

Specifies the number of offsite volumes to reclaim space from. You can specify an integer 0 - 99999. A value of zero means that none of the offsite volumes are reclaimed.

**Tip:** To determine the value for the **OFFSITERECLAIMLIMIT**, use the statistical information in the message that is issued at the end of the offsite volume reclamation operation. 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 are reclaimed when the reclamation runs. If you specify a value of 2, only VOL1 and VOL2 are reclaimed when the reclamation runs. If you specify a value of 1, only VOL1 is 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 0 - 100000000. By allowing the server to request scratch volumes as needed, you avoid having to define each volume to be used.

The value that is 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. An administrator can query the server for empty, offsite scratch volumes and return them 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 use 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 is not 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 0 - 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.

**Tip:** Use this parameter to 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 that are specified for this parameter must be the same as the number specified for the **SET DRMDBBACKUPEXPIREDAYS** command.

**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 ("").

**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 that is stored in your storage hierarchy. You can specify the following values:

**Yes**

Specifies that data is stored containing CRC information, allowing for audit volume processing to validate storage pool data. This mode impacts performance because more processing is required to calculate and compare CRC values between the storage pool and the server.

**No** Specifies that data is stored without CRC information.

**Tip:** For storage pools that are associated with the 3592, LTO, or ECARTRIDGE device type, logical block protection provides better protection against data corruption than CRC validation for a storage pool. If you specify CRC validation for a storage pool, data is validated only during volume auditing operations. Errors are identified after data is written to tape.

To enable logical block protection, specify a value of READWRITE for the **LBPROTECT** parameter on the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands for the 3592, LTO, or ECARTRIDGE device types. Logical block protection is supported only on the following types of drives and media:

- IBM LTO5 and later.
- IBM 3592 Generation 3 drives and later with 3592 Generation 2 media and later.
- Oracle StorageTek T10000C and T10000D drives.

**DEDuplicate**

Specifies whether the data that is stored in this storage pool is 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 server-side duplicate identification. This parameter is optional and is valid only for storage pools that are defined with a FILE device class. Enter a value 1 - 50.

**Remember:** 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, becomes idle, and then processes five more files, then the total number of files processed is nine. Processes end only when canceled or when the number of duplicate-identification processes for the storage pool is changed to a value less than the number currently specified.

**Example: Update an active data pool**

Update the active-data pool that is 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 a virtual file space mapping definition.

**Restriction:** You cannot use the **UPDATE VIRTUALFSMAPPING** command to update a virtual file space mapping for an EMC Celerra or EMC VNX NAS device. You must use the **DEFINE VIRTUALFSMAPPING** command.

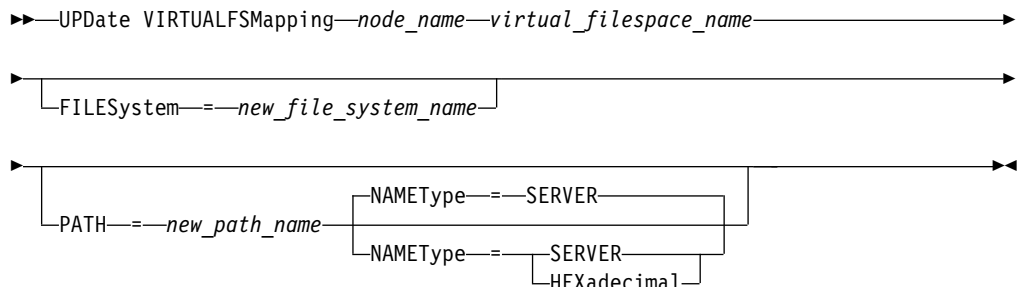
The NAS device needs an associated data mover definition because when the server updates a virtual file space mapping, the server contacts the NAS device to validate the virtual file system and file system name.

### 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



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 447. Commands related to UPDATE VIRTUALFSMAPPING*

| <b>Command</b>          | <b>Description</b>                   |
|-------------------------|--------------------------------------|
| DEFINE VIRTUALFSMAPPING | Define a virtual file space mapping. |
| DELETE VIRTUALFSMAPPING | Delete a virtual file space mapping. |
| QUERY VIRTUALFSMAPPING  | Query a virtual file space mapping.  |



**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 448. Commands related to UPDATE VOLHISTORY*

| <b>Command</b>    | <b>Description</b>                                                                    |
|-------------------|---------------------------------------------------------------------------------------|
| 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 onsite and offsite.                                                   |
| 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)

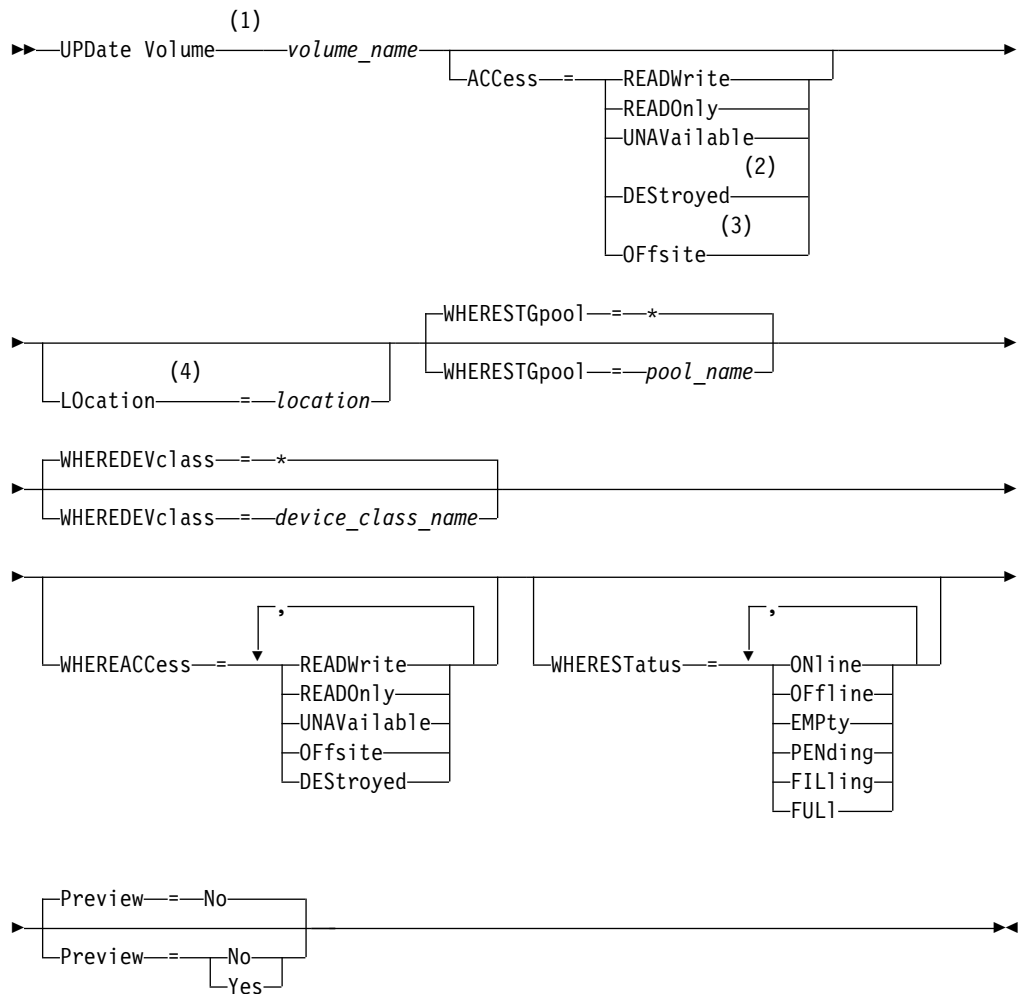
Use this command to change the access mode for one or more volumes in storage pools.

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 STGPPOOL** 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 or active-data storage pool volume is at an offsite location from which it cannot be mounted. Only volumes in copy or active-data storage pools can have the access mode of OFFSITE.

Use this mode to help you track volumes that you move to offsite locations.

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.

**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 449. 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 1443
- “VALIDATE POLICYSET (Verify a policy set)” on page 1445
- “VALIDATE REPLICATION (Validate replication for a client node)” on page 1447
- “VALIDATE REPLPOLICY (Verify the policies on the target replication server)” on page 1452
- “VALIDATE REPLICATION (Validate replication for a client node)” on page 1447



## 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 | Class | Destination Name | LAN-Free capable? | Explanation                                                                                                  |
|-----------|---------------|-----------|-----------------|-------|------------------|-------------------|--------------------------------------------------------------------------------------------------------------|
| TIGER     | AIX_STA1      | BACKUP    | STANDARD        |       | OUTPOOL          | NO                | No available online paths.                                                                                   |
| TIGER     | AIX_STA1      | BACKUP    | STANDARD        |       | PRIMARY          | NO                | Destination storage pool is configured for simultaneous write.                                               |
| TIGER     | AIX_STA1      | BACKUP    | STANDARD        |       | SHRPOOL          | YES               |                                                                                                              |
| TIGER     | AIX_STA1      | BACKUP    | NOARCH          |       | LFFILE           | NO                | Storage pool contains data deduplicated by clients, and is not accessible by storage agents V6.1 or earlier. |
| TIGER     | AIX_STA1      | ARCHIVE   | STANDARD        |       | OUTPOOL          | NO                | No available online paths.                                                                                   |
| TIGER     | AIX_STA1      | ARCHIVE   | STANDARD        |       | PRIMARY          | NO                | Destination storage pool is configured for simultaneous write.                                               |
| TIGER     | AIX_STA1      | ARCHIVE   | STANDARD        |       | SHRPOOL          | YES               |                                                                                                              |

### Related commands

Table 450. 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. |

Table 450. Commands related to **VALIDATE LANFREE** (continued)

| <b>Command</b>  | <b>Description</b>                                                                      |
|-----------------|-----------------------------------------------------------------------------------------|
| QUERY DRIVE     | Displays information about drives.                                                      |
| QUERY LIBRARY   | Displays information about one or more libraries.                                       |
| 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 451. 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.                                                      |

## VALIDATE REPLICATION (Validate replication for a client node)

Use this command to identify the replication rules that apply to file spaces in client nodes that are configured for replication. You can also use this command to verify that the source replication server can communicate with the target replication server.

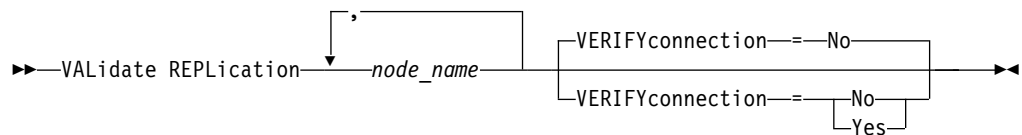
Before you begin replication processing, use the **VALIDATE REPLICATION** command to determine whether your replication configuration is correct.

Issue this command on the server that acts as a source for replicated data.

### Privilege class

To issue this command, you must have system privilege.

### Syntax



### Parameters

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

Specifies the name of the client node whose file spaces you want to display. To specify multiple client node names, separate the names with commas and no intervening spaces. You can use wildcard characters to specify names.

Information is displayed only for client nodes that are either enabled or disabled for replication. The replication mode must be SEND. To determine whether a client node is enabled or disabled for replication and its mode, issue the **QUERY NODE** command. Look for values in the Replication State and Replication Mode fields.

#### **VERIFYconnection**

Specifies whether to check the connection to a target replication server. The version of the target replication server is also checked to verify that it is Version 6.3 or later. This parameter is optional. The default is NO. You can specify one of the following values:

**No** The connection and version of the target replication server are not checked.

**Yes**

The connection and version of the target replication server are checked.

### Example: Validate replication for a client node

The name of the client node is NODE1. Verify the connection status between the source and the target replication servers.

```
validate replication node1 verifyconnection=yes
```

```

Node Name: NODE1
Filespace Name: \\node1\c$
FSID: 1
Type: Bkup
Controlling Replication Rule: ACTIVE_DATA
Replication Rule Level: System Level
Server Name: DRSRV
Connection Status: Valid Connection

Node Name: NODE1
Filespace Name: \\node1\c$
FSID: 1
Type: Arch
Controlling Replication Rule: ALL_DATA_HIGH_PRIORITY
Replication Rule Level: Node Level
Server Name: DRSRV
Connection Status: Valid Connection

Node Name: NODE1
Filespace Name: \\node1\c$
FSID: 1
Type: SpMg
Controlling Replication Rule: ALL_DATA
Replication Rule Level: System Level
Server Name: DRSRV
Connection Status: Valid Connection

```

Output is displayed for all data types regardless of whether a file space contains the data types. For example, if a file space contains only backup and archive data, the output of the **VALIDATE REPLICATION** command also contains information that would be relevant to space-managed data.

## Field descriptions

### Node Name

The node that owns the replicated data.

### Filespace Name

The name of the file space that belongs to the node.

File space names can be in a different code page or locale than the server. If they are, the names in the Operations Center and the administrative command-line interface might not be displayed correctly. Data is backed up and can be restored normally, but the file space name or file name might be displayed with a combination of invalid characters or blank spaces.

If the file space name is Unicode-enabled, Tivoli Storage Manager converts the name to the server code page for display. The success of the conversion depends on the operating system, the characters in the name, and the server code page. Conversion can be incomplete if the string includes characters that are not available in the server code page or if the server cannot access system conversion routines. If the conversion is incomplete, the name might contain question marks, blanks, unprintable characters, or ellipses (...).

**FSID** The file space identifier for the file space. The server assigns a unique FSID when a file space is first stored on the server.

**Type** The type of data. The following values are possible:

**Arch** Archive data

**Bkup** Backup data

**SpMg** Data that was migrated by a Tivoli Storage Manager for Space Management client.

### **Controlling Replication Rule**

The name of the replication rule that controls replication for a data type in a file space. To determine whether the controlling rule is a file space rule, a client rule, or a server rule, check the Replication Rule Level field.

### **Replication Rule Level**

The level of the controlling rule in the replication-rule hierarchy. The following values are possible:

#### **Filespace**

The controlling rule is assigned to a data type in the file space.

**Node** The controlling rule is assigned to a data type for a client node.

**Server** The controlling rule is assigned to a data type for all file spaces in all client nodes that are configured for replication.

### **Server Name**

The name of the target replication server to be queried.

### **Connection Status**

The connection status between the source and the target replication server. The following values are possible:

#### **Valid Connection**

Communication with the target replication server was successful, and the target replication server is a V6.3 server.

#### **Target Server Not Set**

The target replication server is not set. To set the target replication server, issue the **SET REPLSERVER** command.

#### **Communication Failure**

The source replication server was unable to contact the target replication server. Examine the activity log for error messages about failed communications. Consider the following possible causes:

- The replication configuration on the source replication server is not valid. One or more of the following problems might exist:
  - The server definition for the target replication server is incorrect.
  - If the target replication-server definition was deleted and redefined, issue the **PING SERVER** command to test the connection between the source and the target replication server. If the **PING SERVER** command is successful, issue the **UPDATE SERVER** command and specify **FORCESYNC=YES** to reset the server verification keys.
  - The server name, server low-level address, server high-level address, and server password do not match the values that are specified in the server definition on the target replication server.
- The replication configuration on the target replication server is not valid. One or more of the following problems might exist:
  - The version of the target replication server is earlier than V6.3.

- The server definition for the source replication server is incorrect.
- The server name, server low-level address, server high-level address, and server password do not match the values that are specified in the server definition on the source replication server.
- Network communications are unavailable. To test the connection between the source and target server, issue the **PING SERVER** command.
- The target replication server is unavailable.
- Sessions between the source and the target replication servers are disabled. To verify the status of sessions, issue the **QUERY STATUS** command.

### Replication Suspended

Replication processing is suspended when you restore the database on the source replication server or you disable replication processing on this server by issuing the **DISABLE REPLICATION** command.

## Related commands

Table 452. Commands related to **VALIDATE REPLICATION**

| Command               | Description                                                                                     |
|-----------------------|-------------------------------------------------------------------------------------------------|
| DISABLE REPLICATION   | Prevents outbound replication processing on a server.                                           |
| ENABLE REPLICATION    | Allows outbound replication processing on a server.                                             |
| ENABLE SESSIONS       | Resumes server activity following the <b>DISABLE</b> command or the <b>ACCEPT DATE</b> command. |
| 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 REPLRULE        | Displays information about node replication rules.                                              |
| QUERY SERVER          | Displays information about servers.                                                             |
| QUERY STATUS          | Displays the settings of server parameters, such as those selected by the <b>SET</b> commands.  |
| REPLICATE NODE        | Replicates data in file spaces that belong to a client node.                                    |
| SET ARREPLRULEDEFAULT | Specifies the server node-replication rule for archive data.                                    |
| SET BKREPLRULEDEFAULT | Specifies the server node-replication rule for backup data.                                     |
| SET REPLSERVER        | Specifies a target replication server.                                                          |
| SET SPREPLRULEDEFAULT | Specifies the server node-replication rule for space-managed data.                              |
| UPDATE FILESPACE      | Changes file-space node-replication rules.                                                      |



*Table 452. Commands related to VALIDATE REPLICATION (continued)*

| <b>Command</b>  | <b>Description</b>                                             |
|-----------------|----------------------------------------------------------------|
| UPDATE NODE     | Changes the attributes that are associated with a client node. |
| UPDATE REPLRULE | Enables or disables replication rules.                         |
| UPDATE SERVER   | Updates information about a server.                            |

## VALIDATE REPLPOLICY (Verify the policies on the target replication server)

Use this command to compare the policies for client nodes on the source replication server with the same policies on the target replication server where the client node data is being replicated.

The command displays the differences between these policies so that you can verify that any differences between the policies on the source and target replication servers are intended or you can modify the policies on the target replication server.

Ensure that Tivoli Storage Manager, Version 7.1.1 or later, is installed on the source and target replication servers before you issue this command. Issue this command on the source replication server.

### Privilege class

Any administrator can issue this command.

### Syntax

```
➤ VALidate REPLPolicy server_name
```

### Parameters

*server\_name*

Specifies the name of the target replication server that has policies you want to verify. This parameter is optional. If you do not specify this parameter, the command sets the default replication server as the target replication server.

### Example: Display the differences between the replication policies on a source and target replication server

To display the differences between the policies on the source replication server and the policies on the target replication server, CVTCVS\_LXS\_SRV2, where the client data is replicated, issue the following command on the source replication server:

```
VALIDATE REPLPOLICY CVTCVS_LXS_SRV2
```

| Policy domain name on this server             | Policy domain name on target server | Target server name       |
|-----------------------------------------------|-------------------------------------|--------------------------|
| -----<br>STANDARD                             | -----<br>STANDARD                   | -----<br>CVTCVS_LXS_SRV2 |
| Differences in policy set:<br>Change detected | Source server value                 | Target server value      |
| -----                                         | -----                               | -----                    |
| Mgmt class only on target                     | Not applicable                      | STANDARD2                |
| Mgmt Class only on source                     | STANDARD1                           | Not applicable           |
| Differences in backup copy group              | STANDARD in management class        | STANDARD                 |
| Change detected                               | Source server value                 | Target server value      |
| -----                                         | -----                               | -----                    |
| Versions data exists                          | 2                                   | 20                       |
| Affected nodes                                |                                     |                          |
| -----                                         |                                     |                          |
| NODE1,NODE2,NODE3,NODE4,NODE5                 |                                     |                          |

## Field descriptions

### Policy domain name on this server

Specifies the policy domain name on the source replication server where the command is issued.

### Policy domain name on target server

Specifies the policy domain name on the target replication server.

### Target server name

Specifies the name of the target replication server.

### Differences in policy set:

Specifies the differences between the policies that are defined on the source and target replication servers. The differences between the policies are listed under the following fields:

#### Change detected

Specifies the list of policy items that are different between the source and target replication servers.

#### Source server value

Specifies the value for the policy item on the source replication server.

#### Target server value

Specifies the value for the policy item on the target replication server.

### Differences in backup copy group <backup\_copy\_group\_name> in default management class OR Differences in archive copy group <archive\_copy\_group\_name> in default management class

Specifies the differences between the backup copy group or the archive copy group in the management class. The differences are listed under the following fields:

#### Change Detected

Specifies the list of copy group fields that are different.

#### Source server value

Specifies the value in the copy group field on the source replication server.

#### Target server value

Specifies the value in the copy group field on the target replication server.

### Affected nodes

Specifies the names of all the client nodes that are affected by the changes that are shown in this output.

## Related commands

Table 453. Commands related to `VALIDATE REPLPOLICY`

| Command                             | Description                                                                     |
|-------------------------------------|---------------------------------------------------------------------------------|
| <code>VALIDATE REPLICATION</code>   | Verifies replication for file spaces and data types.                            |
| <code>QUERY REPLSERVER</code>       | Displays information about replicating servers.                                 |
| <code>SET DISSIMILARPOLICIES</code> | Enable the policies on the target replication server to manage replicated data. |

*Table 453. Commands related to VALIDATE REPLPOLICY (continued)*

| <b>Command</b>  | <b>Description</b>                             |
|-----------------|------------------------------------------------|
| QUERY DOMAIN    | Displays information about policy domains.     |
| QUERY POLICYSET | Displays information about policy sets.        |
| QUERY COPYGROUP | Displays the attributes of a copy group.       |
| QUERY MGMTCLASS | Displays information about management classes. |

---

## VARY (Bring a random access volume online or offline)

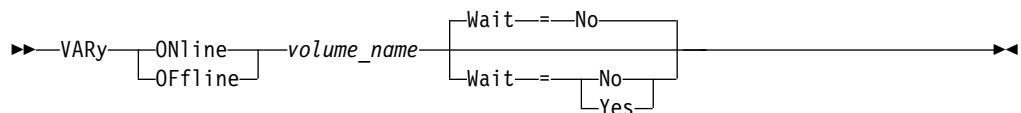
Use this command to make a random access storage pool volume online or offline to the server.

### 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.

### Example: Bring volume online

Make volume /storage/pool001 available to the server for use as a storage pool volume.

```
vary online /storage/pool001
```

## Related commands

*Table 454. Commands related to VARY*

| <b>Command</b> | <b>Description</b>                                                       |
|----------------|--------------------------------------------------------------------------|
| 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.                         |

---

## 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 in the server instance directory

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.

#### About this task

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 1190 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.

### Server communication options

You can use server options to specify server communication methods and their characteristics.

Table 455. Communication options

| Option                | Description                                                                                                                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADMINCOMMTIMEOUT      | The amount of time that the server waits for an administrative client message during an operation that causes a database update                                                                  |
| ADMINIDLETIMEOUT      | The amount of time an administrative client session can be idle                                                                                                                                  |
| ADMINONCLIENTPORT     | The port that determines whether administrative sessions can use the port specified in the TCPPORT option                                                                                        |
| COMMMETHOD            | The server communication method                                                                                                                                                                  |
| DBMTCPPORT            | The port number on which the TCP/IP communication driver for the database manager waits for client session requests                                                                              |
| DNSLOOKUP             | Control of use of Domain Name Services to lookup names of systems contacting the server                                                                                                          |
| LDAPCACHEDURATION     | Determines the amount of time that authentication sessions, to the same node or administrator, are skipped. You might see a slight performance boost when skipping sessions.                     |
| LDAPURL               | Specifies the LDAP directory server. Each setting must have the LDAP directory server name, a port number, and the base distinguished name of the namespace or suffix that the server maintains. |
| NDMPCONTROLPORT       | The internal communications port used for certain Network Data Management Protocol (NDMP) operations                                                                                             |
| NDMPENABLEKEEPALIVE   | The TCP keepalive mechanism                                                                                                                                                                      |
| 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                                                                                                |



Table 455. Communication options (continued)

| Option              | Description                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SNMPSUBAGENTHOST    | The location of the Tivoli Storage Manager SNMP subagent                                                                                                                                                                                                                                                                                                           |
| SNMPSUBAGENTPORT    | The port address of the Tivoli Storage Manager SNMP subagent                                                                                                                                                                                                                                                                                                       |
| SSLDISABLELEGACYTLS | Specifies whether to use protocols earlier than Transport Layer Security (TLS) 1.2 for Secure Sockets Layer (SSL) sessions between the server and the backup-archive client or storage agent                                                                                                                                                                       |
| SSLFIPSMODE         | Specifies whether the Federal Information Processing Standards (FIPS) mode is in effect for Secure Sockets Layer (SSL)                                                                                                                                                                                                                                             |
| 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         | <p>The SSL-only port number on which the server's TCP/IP communication driver waits for requests for SSL-enabled sessions from the following sources:</p> <ul style="list-style-type: none"> <li>• Command line backup-archive client</li> <li>• Backup-archive GUI</li> <li>• Administrative client</li> <li>• Application programming interface (API)</li> </ul> |
| SSLTLS12            | Controls Transport Layer Security (TLS) 1.2, an SSL protocol that is available for use with sessions between the server and the backup-archive client or storage agent                                                                                                                                                                                             |
| 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                                                                                                                                                                                                                                                                                                                         |
| TCPWINDOWSIZE       | The client node TCP/IP sliding window                                                                                                                                                                                                                                                                                                                              |

## Server storage options

Tivoli Storage Manager provides a number of options that you can specify to configure certain device and server storage operations.

Table 456. 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 3590 drive in recovering from a lost or corrupted Vital Cartridge Records (VCR) condition.                                                                           |
| CHECKTAPEPOS        | Specifies whether the server validates data position on tape.                                                                                                                                                    |
| CLIENTDEDUPTXNLIMIT | Specifies the maximum size of a transaction when client-side deduplicated data is backed up or archived.                                                                                                         |
| DEDUPREQUIRESBACKUP | Specifies whether volumes in primary sequential-access storage pools that are set up for data deduplication can be reclaimed and whether duplicate data can be discarded before the storage pools are backed up. |
| DEDUPTIER2FILESIZE  | File size at which Tier 2 processing is used for data deduplication.                                                                                                                                             |
| DEDUPTIER3FILESIZE  | File size at which Tier 3 processing is used for data deduplication.                                                                                                                                             |
| 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.                                                                        |
| ENABLENASDEDUP      | Specifies whether the server deduplicates data that is stored by a NetApp network-attached storage (NAS) file server.                                                                                            |
| NUMOPENVOLSALLOWED  | The number of input FILE volumes in a deduplicated storage pool that can be open at one time.                                                                                                                    |
| 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 the resource.                                                                                                |
| RETENTIONEXTENSION  | The number of days to extend the retention date of a SnapLock volume.                                                                                                                                            |
| SEARCHMPQUEUE       | The order in which the server satisfies requests in the mount queue.                                                                                                                                             |
| SERVERDEDUPTXNLIMIT | Specifies the maximum size of objects that can be deduplicated on the server.                                                                                                                                    |

## Client-server options

You can use server options to control client-server processing.

Table 457. 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                                                       |
| 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 458. 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 459. Database options

| Option                           | Description                                                                                                                                |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| ACTIVELOGDIRECTORY               | 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.                                                                                                        |
| ALLOWREORGINDEX                  | Server-initiated index reorganization.                                                                                                     |
| ALLOWREORGTABLE                  | Server-initiated table reorganization.                                                                                                     |
| ARCHLOGDIRECTORY                 | The directory that the database manager can archive a log file into after all the transactions represented in that log file are completed. |
| ARCHFAILOVERLOGDIRECTORY         | The directory in which the server tries to store archive log files that cannot be stored in the archive log directory.                     |
| DBDIAGLOGSIZE                    | The maximum size of the database manager diagnostic log files.                                                                             |
| DBDIAGPATHFSTHRESHOLD            | The threshold for free space on the file system or disk that contains the database manager diagnostic log files.                           |
| DBMEMPERCENT                     | The percentage of system memory that is dedicated to the database.                                                                         |
| "DISABLEREORGTABLE" on page 1500 | Disables table reorganization for specific tables.                                                                                         |

Table 459. Database options (continued)

| Option             | Description                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------|
| FSUSEDTHRESHOLD    | The percentage of the file system that can be used by the database before an alert message is issued. |
| MIRRORLOGDIRECTORY | The directory for mirroring the active log path.                                                      |
| REORGBEGINTIME     | The earliest time that the Tivoli Storage Manager server can start a table or index reorganization.   |
| REORGDURATION      | The interval during which server-initiated table or index reorganization can start.                   |

## Data transfer options

You can use server options to control how Tivoli Storage Manager groups and transfers data.

Table 460. 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 |
| NDMPPREFDATAINTERFACE | The IP address associated with the interface in which the server receives all Network Data Management Protocol (NDMP) backup data |
| REPLBATCHSIZE         | The number of files that are to be replicated in a batch, within the same server transaction                                      |
| REPLSIZETHRESH        | The threshold for the amount of data replicated as a batch, within the same server transaction                                    |
| 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 461. 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 462. 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)                                          |
| FILETEXTTEXT         | 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 463. 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                                              |
| LDAPURL           | Specifies the LDAP directory server. Each setting must have the LDAP directory server name, a port number, and the base distinguished name of the namespace or suffix that the server maintains. |
| 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                                                             |
| SHREDDING         | Specifies whether shredding of deleted sensitive data is done automatically or manually                                                                                                          |

## Miscellaneous options

You can use a variety of miscellaneous server options to customize Tivoli Storage Manager.

Table 464. Miscellaneous options

| Option          | Description                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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                                                                                                         |
| FFDCLOGNAME     | The name for the first failure data capture (FFDC) log                                                                                                                |
| FFDCMAXLOGSIZE  | The maximum size of the first failure data capture (FFDC) log                                                                                                         |
| 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

▶▶—3494SHARED—Yes  
No

### 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

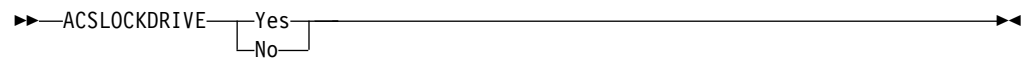


---

## 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
```

---

## ACTIVELOGDIRECTORY

The ACTIVELOGDIRECTORY 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 1594 for guidance on this option.

### Syntax

▶▶—ACTIVELOGDirectory—*dir\_name*—————▶▶

### Parameters

*dir\_name*

Specifies a fully qualified directory name. The directory must 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

activelogdirectory /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 1594 for guidance on this option.

### Syntax

►► ACTIVELOGSize 16GB  
megabytes ◀◀

### Parameters

*megabytes*

Specifies the size of the active log file in megabytes. The minimum value is 2048 MB (2 GB); the maximum is 524,288 MB (512 GB). If an odd number is specified, the value is rounded up to the next even number. The default is 16,384 MB (16 GB).

The size of an active log file that is used by Tivoli Storage Manager is based on the value of the ACTIVELOGSIZE option. Guidelines for space requirements are in the following table:

*Table 465. How to estimate volume and file space requirements*

| ACTIVELOGSize option value | Reserve this much free space in the active log directory, in addition to the ACTIVELOGSize space |
|----------------------------|--------------------------------------------------------------------------------------------------|
| 2 GB - 128 GB              | 5120 MB                                                                                          |
| 129 GB - 256 GB            | 10240 MB                                                                                         |
| 257 GB - 512 GB            | 20480 MB                                                                                         |

### Examples

activesize 8192

---

## ADMINCOMMTIMEOUT

The ADMINCOMMTIMEOUT option specifies how long the server waits for an expected administrative client message during an operation that causes a database update.

If the length of time exceeds this time-out period, the server ends the session with the administrative client. You may want to increase the time-out value to prevent administrative client sessions from timing out.

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 1190.

### Syntax

►► ADMINCOMMTIMEOUT 60 seconds ◄◄

### Parameters

*seconds*

Specifies the maximum number of seconds that a server waits for an administrative client response. The default value is 60. The minimum value is 1.

### Examples

```
admincommtimeout 60
```

---

## ADMINIDLETIMEOUT

The ADMINIDLETIMEOUT option specifies the amount of time, in minutes, that an administrative client session can be idle before the server cancels the session.

If there is a heavy network load in your environment, you might want to increase the time-out value to prevent administrative clients from timing out. However, 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 1190.

### Syntax

►► ADMINIDLETIMEOUT 15  
minutes ◀◀

### Parameters

*minutes*

Specifies the maximum number of minutes that a server waits for an idle administrative client. The default value is 15 minutes. The minimum value is 1 minute.

### Examples

```
adminidletimeout 20
```

---

## ADMINONCLIENTPORT

The ADMINONCLIENTPORT option defines whether or not the TCPPOINT can be used by administrative sessions. The default is YES.

### Syntax

▶▶—ADMINONCLIENTPORT—YES  
                                  └─NO—┘▶▶

### Parameters

#### YES

If the option is set to YES, or if the TCPPOINT and TCPADMINPORT are the same value (the default), then administrative sessions can use the TCPPOINT.

**NO** If the option is set to NO, and if the TCPADMINPORT value is different than the TCPPOINT value, then administrative sessions cannot use the TCPPOINT.

### Examples

Specify that the TCPPOINT can be used by administrative sessions.

```
adminonclientport 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

aliashalt tsmhalt

---

## ALLOWREORGINDEX

The ALLOWREORGINDEX option specifies whether server-initiated index reorganization is enabled or disabled.

The default is YES.

### Syntax

▶▶—ALLOWREORGINDEX  Yes  No

### Parameters

#### Yes

Specifies that server-initiated index reorganization is enabled.

**No** Specifies that server-initiated index reorganization is disabled.

### Example

Specify that server-initiated index reorganization is enabled.

```
allowreorgindex yes
```

---

## ALLOWREORGTABLE

The ALLOWREORGTABLE option specifies whether server-initiated table reorganization is enabled or disabled.

The default is YES.

### Syntax

▶▶—ALLOWREORGTABLE  Yes  No ▶▶

### Parameters

#### Yes

Specifies that server-initiated table reorganization is enabled.

**No** Specifies that server-initiated table reorganization is disabled.

### Examples

Specify that server-initiated table reorganization is disabled.

```
allowreorgtable no
```

---

## ARCHFAILOVERLOGDIRECTORY

The ARCHFAILOVERLOGDIRECTORY 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 DSMSERV FORMAT command is run. Typically the directory does not need to be changed.

### Syntax

▶▶—ARCHFailoverlogdirectory—*dir\_name*————▶▶

### Parameters

*dir\_name*

Specifies a fully qualified directory name. The maximum number of characters is 175.

### Examples

archfailoverlogdirectory /tsm/archfailoverlog

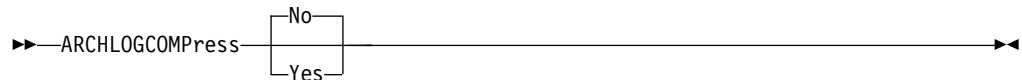
---

## ARCHLOGCOMPRESS

You can enable or disable compression of archive logs on the Tivoli Storage Manager server. By compressing the archive logs, you reduce the amount of space that is required for storage.

The ARCHLOGCOMPRESS server option specifies whether log files that are written to the archive directory for logs are compressed.

### Syntax



### Parameters

**No** Specifies that log files that are written to the archive log directory are not compressed. The default is No.

**Yes**

Specifies that log files that are written to the archive log directory are compressed.

**Restriction:** Use caution when you enable the ARCHLOGCOMPRESS server option on systems with sustained high volume usage and heavy workloads. Enabling this option in this system environment can cause delays in archiving log files from the active log file system to the archive log file system. This delay can cause the active log file system to run out of space. Be sure to monitor the available space in the active log file system after archive log compression is enabled. If the active log directory file system usage nears out of space conditions, the ARCHLOGCOMPRESS server option must be disabled. You can use the **SETOPT** command to disable archive log compression immediately without halting the server.

### Example

To enable compression of log files that are written to the archive log directory, specify the following option:

```
archlogcompress yes
```

---

## ARCHLOGDIRECTORY

The ARCHLOGDIRECTORY 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 DSMSERV FORMAT command is run.

### Syntax

►►—ARCHLOGDirectory—*dir\_name*—————►►

### Parameters

*dir\_name*

Specifies a fully qualified directory name. The maximum number of characters is 175.

### Examples

archlogdirectory /tsm/archlog

---

## ARCHLOGUSEDTHRESHOLD

The ARCHLOGUSEDTHRESHOLD option specifies when to start an automatic database backup in relation to the percentage of archive log file space used. The default is 80 percent.

The **ARCHLOGUSEDTHRESHOLD** option prevents frequent automatic backups. For example, if the archive log file directory resides on a file system or drive that is 400 GB, a database backup is triggered if there is less than 80 GB of free space. Repeated database backups might cause the server to use an excessive amount of scratch tapes.

### Syntax

►► ARCHLOGUSEDTHRESHOLD 80  
*value* ◄◄

### Parameters

*value*

The percentage of archive log file space used before an automatic backup starts.

Specify to start an automatic backup when 90 percent of archive log file space is used.

```
archlogusedthreshold 90
```

---

## ASSISTVCRRECOVERY

The ASSISTVCRRECOVERY option specifies whether Tivoli Storage Manager assists an IBM 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 might be some small effect on performance because the drive cannot complete 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
```

---

## CHECKTAPEPOS

The CHECKTAPEPOS option specifies whether the Tivoli Storage Manager server validates the position of data blocks on tape.

The CHECKTAPEPOS option applies only to operations that use tape drives. It does not apply to non-tape, sequential-access device classes such as FILE. If the server information about position does not match the position that is detected by the drive, an error message is displayed, the transaction is rolled back, and the data is not committed to the database.

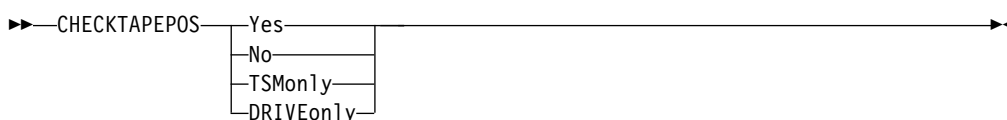
Using the CHECKTAPEPOS option, you can enable append-only mode for IBM LTO Generation 5 and later drives, and for any drives that support this feature. When it is enabled, the drive issues an error after it receives instructions to overwrite any data on the currently mounted volume. The Tivoli Storage Manager server repositions the tape to the correct block and continues writing data. Append-only mode provides added protection by preventing most data overwrite situations. If you are using a drive that supports this feature, you can validate data position on tape by using both Tivoli Storage Manager and the drive or you can enable one or the other.

**Note:** When you use SAN Tape acceleration functions in the fabric, set CHECKTAPEPOS to DRIVEonly or No to avoid false positive positioning errors. The Tivoli Storage Manager CHECKTAPEPOS server option does not require an append-only capable drive.

Changes to the CHECKTAPEPOS option affect mounts only after the update to the drive is complete.

The default is YES.

### Syntax



### Parameters

#### Yes

Specifies that the Tivoli Storage Manager server validates data position on tape. For drives that support append-only mode, this parameter specifies that Tivoli Storage Manager enables the drive to also validate the data position during each WRITE operation to prevent data overwrite. Yes is the default.

**No** Specifies that all data position validation is turned off.

#### TSMonly

Specifies that the Tivoli Storage Manager server validates data position on tape. The server does not use append-only mode even if the drive supports the feature

#### DRIVEonly

Specifies that the Tivoli Storage Manager server enables append-only mode for drives that support this feature. The server does not validate the data position on tape.

## Example

Validate data position on tape and enable append-only mode for a supported drive:

```
checktapepos yes
```

---

## CLIENTDEDUPTXNLIMIT

The CLIENTDEDUPTXNLIMIT option specifies the maximum size of a transaction when client-side deduplicated data is backed up or archived.

When you use client-side deduplication for large objects, intensive database activity can result from long-running transactions that are required to update the database. High levels of database activity can produce the following symptoms:

- Reduced throughput for client backup and archive operations
- Resource contention resulting from concurrent server operations
- Excessive recovery log activity

The extent to which these symptoms occur depends on the number and size of objects being stored using client-side data deduplication, the intensity and type of concurrent operations taking place on the Tivoli Storage Manager server, and the Tivoli Storage Manager server configuration.

With the CLIENTDEDUPTXNLIMIT server option, you can specify a maximum size, in gigabytes, for transactions when client-side deduplicated data is backed up or archived. If an object or set of objects in a single transaction exceeds the limit specified by CLIENTDEDUPTXNLIMIT, the objects are not deduplicated by the client, and the transaction can fail. You can specify a value 32 - 102400 GB. The default value is 5120 GB.

If an object or set of objects in a single transaction exceeds the limit specified by CLIENTDEDUPTXNLIMIT, the objects or set of objects is not deduplicated by the client. However, the objects are sent to the server. These objects can be deduplicated on the server, depending on whether the destination storage pool is configured for data deduplication and on the value of the SERVERDEDUPTXNLIMIT option. Objects in a deduplication-enabled storage pool that are less than the value of the SERVERDEDUPTXNLIMIT are deduplicated by a server duplicate-identification process.

The appropriate value for this option depends on the Tivoli Storage Manager server configuration and concurrent server activity. You can specify a high value for this option if you minimize resource contention. To minimize resource contention, perform operations, such as backup, archive, duplicate identification (the **IDENTIFY DUPLICATES** command), and reclamation, at different times.

To update this server option without stopping and restarting the server, use the **SETOPT** command.

### Syntax

►► CLIENTDEDUPTXNLIMIT *gigabytes* 5120 ◀◀

### Parameters

*gigabytes*

Specifies the maximum size, in gigabytes, of objects that can be backed up or archived using client-side data deduplication. You can specify a value 32 - 102400. The default value is 5120.

## Examples

Disable client-side data deduplication for all objects over 80 GB:

```
clientdeduptxnlimit 80
```

---

## COMMMETHOD

The COMMMETHOD option specifies a communication method to be used by the server.

You can configure the server to use multiple communication methods. The more commonly used are the TCPIP, V6TCPIP, and SHAREDMEM communication methods. To specify multiple communication methods, enable each method by adding a COMMMETHOD stanza to the dsmserv.opt options file.

**Important:** When you enable a communication method, you must also add the options that are specific to the communication method to the 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.

#### SHAREDMEM

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 must be configured to support the shared memory communication method, and they must be installed on the same computer.

#### SNMP

Specifies the SNMP communication method option.

#### TCPIP

Specifies the TCP/IP communication method option. This option 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 V4 or TCP/IP V6 if this option is specified.

### Examples

Example of specifying multiple communication methods to be used by the server (TCP/IP and TCP/IP Version 6):

```
commethod tcpip  
commethod 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.

The **COMMTIMEOUT** server option is used for non-administrative sessions. See the **ADMINCOMMTIMEOUT** option for administrative client sessions.

You can update this server option without stopping and restarting the server by using the **SETOPT** command.

### 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
```



---

## DBDIAGLOGSIZE

This option helps to control the amount of space that is used by diagnostic log files.

The database manager uses diagnostic log files to log messages. You must control the size of the log files so that they do not fill the file system. Use the **DBDIAGLOGSIZE** option to set the amount of space that is used by the log files.

If you set a value in the range 2 - 9999, a maximum of 10 rotating diagnostic log files are retained. Each file name indicates the order in which the file was created. After a file is full, the next file is created. When the 10th file is full, the oldest file is deleted, and a new file is created. The following example shows how the rotating log files might look:

db2diag.14.log, db2diag.15.log, ... , db2diag.22.log, db2diag.23.log

When db2diag.23.log is full, db2diag.14.log is deleted, and db2diag.24.log is created.

The server checks the file space that contains the diagnostic log files every hour. Messages are displayed every 12 hours if either of the following conditions occur:

- The available space in the file system where the diagnostic log files are located is less than 20% of the total file system space.
- The available space in the file system where the server instance directory is located is less than 1 GB.

If you specify a value of 0, only one log file, db2diag.log, is used for all diagnostic messages. No limits are imposed on the size of the log file.

**Restriction:** You must monitor the size of the diagnostic log files to ensure that they do not use all the available space in the file system. If there is not enough available space, the server might fail to respond.

### Syntax

►► DBDIAGLOGSIZE 1024 *megabytes* ◀◀

### Parameters

*megabytes*

Specifies the amount of space that is used by diagnostic log files in megabytes. Specify a value in the range 2 - 9999, or a value of 0. The default value is 1024.

If you specify a value in the range 2 - 9999, rotating log files are used, and the value specifies the total size in megabytes of all 10 log files. The value is reset to 1024 whenever the server is restarted.

If you specify a value of 0, one log file is used, and no limits are imposed on the size of the log file.

If you want to archive messages, specify a value of 0 to ensure that the db2diag.log file can use all the available space without using rotating log files.

After you set the value of the **megabytes** parameter to 0 by using the **DBDIAGLOGSIZE** option, messages are initially written to rotating log files. After the server is restarted, messages are written to the db2diag.log file.

**Tip:** If you specify a value in the range 2 - 9999 by using the server options file, `dsmserv.opt`, the value is not reset automatically at server startup. The value remains the same until it is changed or removed from the `dsmserv.opt` file, by using the **SETOPT** command.

**Example: Specify a maximum size of 5120 megabytes**

Specify the size of the diagnostic log files as 5120 megabytes (5 GB):

```
dbdiaglogsize 5120
```

**Example: Archive messages in a single log file**

Archive messages by specifying that the messages are written to the `db2diag.log` file:

```
dbdiaglogsize 0
```

---

## DBDIAGPATHFSTHRESHOLD

The DBDIAGPATHFSTHRESHOLD option specifies the threshold for free space on the file system or disk that contains the db2diag.log file.

When the amount of free space is equal to or less than the specified threshold, the ANR1545W error message is shown. By default, the message is shown when the file system or disk has 20% or less of free disk space.

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 1190.

### Syntax

►►—DBDIAGPATHFSTHreshold—*percent*—————►►

### Parameter

*percent*

Specifies the percentage of available space in the file system. Valid values are in the range 0 - 100. The default is 20.

**Tip:** For best results, do not set a low or high value for the **percent** parameter. A low value might cause the file system to become full before you can correct the issue. A full file system might corrupt the server database. A high value might result in many ANR1545W messages in the server activity log.

### Example

Set the threshold value to 10%.

```
setopt DBDIAGPATHFSTH 10
```

---

## 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 that is between 75 percent and 95 percent of system RAM. The default value is AUTO.

### Examples

dbmempercent 50

---

## DBMTCPPORT

The DBMTCPPORT option specifies the port number on which the TCP/IP communication driver for the database manager waits for requests for client sessions.

The specified port number must be reserved for use by the database manager.

By default, the Tivoli Storage Manager server uses interprocess communications (IPC) to establish connections for the first two connection pools, with a maximum of 480 connections for each pool. After the first 960 connections are established, the Tivoli Storage Manager server uses TCP/IP for any additional connections.

### Syntax

▶▶—DBMTCPPort—*port\_number*—————▶▶

### Parameters

*port\_number*

Specifies the number of the TCP/IP port on which the database manager waits for communications from the server. Valid values are integers from 1024 to 65535.

The default port number is the value of the server TCPPORT option plus 50,000. For example, if the server TCPPORT option is 1500, the default DBMTCPPORT port number would be 51500.

If the TCPPORT server option is greater than 9999, add the last four digits of its value to 50000. For example, if the TCPPORT option is 11500, 1550 is added to 50000, resulting in a DBMTCPPORT port number of 51500.

### Example

```
dbmtcport 51500
```

---

## DEDUPREQUIRESBACKUP

The DEDUPREQUIRESBACKUP option specifies whether volumes in primary sequential-access storage pools that are set up for data 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 data deduplication. Use the BACKUP STGPOOL 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 data 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 data deduplication has been backed up to a copy storage pool. These checks require more than one BACKUP STGPOOL 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 data 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 data deduplication do not have to be backed up.

```
deduprequiresbackup no
```

---

## DEDUPTIER2FILESIZE

The DEDUPTIER2FILESIZE option specifies at what file size Tivoli Storage Manager begins to use Tier 2 data deduplication.

### Syntax

►►—DEDUPTIER2FILESIZE—*nnn*—◄◄

### Parameters

*nnn*

Specifies the file size, in gigabytes, at which point the Tivoli Storage Manager server begins to use Tier 2 processing for data deduplication. You can specify a value 20 - 9999. The default is 100.

**Note:** If the value specified or defaulted to for this option is greater than the value for the SERVERDEDUPTXNLIMIT option, then this option is ignored for server data deduplication. If the value specified or defaulted to for this option is greater than the value for CLIENTDEDUPTXNLIMIT, then this option is ignored for client data deduplication.

### Examples

```
deduptier2filesize 550
```

---

## DEDUPTIER3FILESIZE

The DEDUPTIER3FILESIZE option specifies at what file size Tivoli Storage Manager begins to use Tier 3 data deduplication.

### Syntax

►—DEDUPTIER3FILESIZE—*nnn*—◄

### Parameters

*nnn*

Specifies the file size, in gigabytes, at which point the Tivoli Storage Manager server begins to use Tier 3 processing for data deduplication. You can specify a value 90 - 9999. The default is 400.

- If the value specified or defaulted to for this option is greater than the value for the SERVERDEDUPTXNLIMIT option, then this option is ignored for server data deduplication.
- If the value specified or defaulted to for this option is greater than the value for CLIENTDEDUPTXNLIMIT, then this option is ignored for client data deduplication.
- If the value specified or defaulted to for this option is less than the value specified or defaulted to for DEDUPTIER2FILESIZE, then the value of DEDUPTIER2FILESIZE is used for this option.

### Examples

```
deduptier3filesize 1150
```



---

## 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
- Server 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

---

## DISABLEREORGTABLE

The DISABLEREORGTABLE option specifies whether online table reorganization is disabled for table names that are specified in the tables list.

To use the DISABLEREORGTABLE option, you must halt the server, update the options file, and then restart the server.

### Syntax

►►—DISABLEREORGTa**ble**—*tablelist*—◀◀

### Parameters

#### **tablelist**

Specifies a list of table names for which table reorganization is disabled. If you do not specify any table names with the option, or if the option is not in the options file, no tables are disabled.

**Restriction:** The following tables are already excluded from table reorganization processing and cannot be specified for this option:

- STAGED\_EXPIRING\_OBJECTS
- STAGED\_OBJECT\_IDS
- BF\_DEREFERENCED\_CHUNKS
- BF\_QUEUED\_CHUNKS

### Example

```
DISABLEREORGTABLE BF_BITFILE_EXTENTS,REPLICATING_OBJECTS
```

---

## 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
```

---

## DISALLOWDES

The DISALLOWDES option specifies whether to allow Data Encryption Standard (DES) for authentication between the server and the backup-archive client.

To prevent the use of Data Encryption Standard (DES), use the DISALLOWDES option.

### Syntax



### Parameters

#### Yes

Specifies that the server rejects any backup-archive clients that attempt to authenticate with DES-based encryption.

**No** Specifies that the server allows authentication with any backup-archive clients that use DES-based encryption. The default is No.

### Examples

Specify that the server rejects any backup-archive clients that attempt to authenticate with DES encryption:

```
disallowdes yes
```

Specify that the server allows authentication with any backup-archive clients that use DES encryption:

```
disallowdes no
```

---

## 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

►►—DISPLAYLFINFO— Yes  
                           No

### 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_VOL:
```

---

## 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

►►—DNSLOOKUP—Yes  
                  └─No—┘

### 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

►—DRIVEACquireretry—Forever—  
                                  Never—  
                                  —*number\_of\_retries*—

### 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
```

---

## ENABLENASDEDUP

The ENABLENASDEDUP server option specifies whether the server deduplicates data that is stored by a network-attached storage (NAS) file server. This option applies only to NetApp file servers.

If the value of this option is NO, the data stored by the file server is skipped during duplicate-identification processing. If the value of this option is YES, the value of the **DEDUPLICATE** parameter in the storage pool definition must be YES.

### Syntax

►►—ENABLENASDEDUP—No  
                                  Yes—►►

### Parameters

#### Yes

Specifies that Tivoli Storage Manager server deduplicates data stored by a NetApp file server.

**No** Specifies that the server does not deduplicate data stored by a NetApp file server.

### Example

Specify that the server deduplicates data stored by a NetApp file server.

```
enablenasdedup yes
```

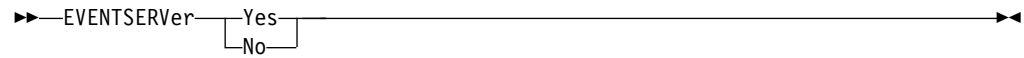


---

## EVENTSERVER

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 1190.

### Syntax

►—EXPINterval 24 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 1190.

### 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
```

---

## FFDCLOGNAME

The FFDCLOGNAME option specifies a name for the first failure data capture (FFDC) log.

The FFDC log file is used to gather diagnostic information about the server. When an error occurs, data about the error is written to the FFDC log file. This information can be provided to IBM Support to help diagnose problems. The FFDC log file is in the server instance directory.

### Syntax

►►—FFDCLOGNAME—<sup>dsdffdc.log</sup>  
*file\_name*—►►

### Parameters

*file\_name*

Specifies a file name for the FFDC log file. The file name can be a fully qualified file name or a file name relative to the server instance directory. The default value is dsmffdc.log.

### Examples

```
ffdclogname /tsminst1/tsdffdc.log  
ffdclogname tsmffdc.log  
ffdclogname c:\tmserv1\tsdffdc.log
```

---

## FFDCMAXLOGSIZE

The FFDCMAXLOGSIZE option specifies the size for the first failure data capture (FFDC) log file.

The FFDC log file is used to gather diagnostic information about the server. When an error occurs, data about the error is written to the FFDC log file. This information can be provided to IBM Support to help diagnose problems.

### Syntax

►►—FFDCMAXLOGSIZE—kilobytes<sup>1024</sup>—►►

### Parameters

*kilobytes*

Specifies the size to which the FFDC log file can grow before wrapping. The minimum value is 500. The maximum value is 2097151. The default value is 1024.

To allow the size of the log file to grow indefinitely, specify a value of -1. To disable the log, specify 0.

### Examples

```
ffdcmaxlogsize 2000
```

---

## FFDCNUMLOGS

The **FFDCNUMLOGS** option specifies the number of log files that can be used for circular logging. The default value is 10.

Circular logging uses a ring of log files to provide recovery from transaction failures and system crashes. For example, when the `dsmffdc.log` file is full, it is renamed to `dsmffdc.log.1`. If a `dsmffdc.log.1` file exists, the `dsmffdc.log.1` file is renamed to `dsmffdc.log.2`. If a `dsmffdc.log.2` exists, the `dsmffdc.log.2` file is renamed to `dsmffdc.log.3`, and so on, until the **FFDCNUMLOGS** value is reached. If there is a log file that is renamed as the **FFDCNUMLOGS** value is reached, that log file is deleted.

The minimum value is 1. The maximum value is 100. The default value is 10.

### Syntax

►► `FFDCNUMLOGS` `value` 10 ◄◄

### Parameters

*value*

Specifies the number of log files that are used for circular logging.

If you specify a value of 1 and the log file size reaches the **FFDCMAXLOGSIZE**, the server continues to write to the log file. Any logging information is overwritten and the server continues to write to the log file.

### Examples

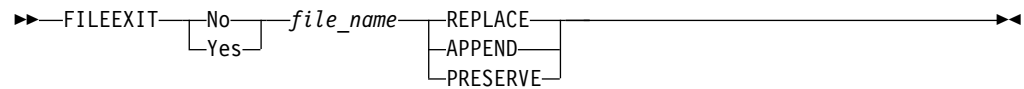
```
ffdcnumlogs 20
```

---

## 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

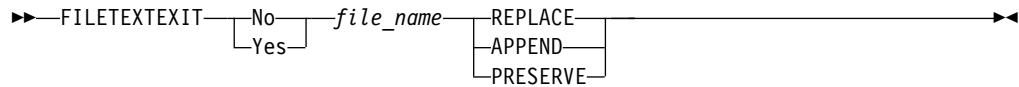
```
fileexit yes /tsm/server/data replace
```

---

## 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
```



---

## FSUSEDTHRESHOLD

The FSUSEDTHRESHOLD option specifies what percentage of the file system can be filled up by the database before an alert message is issued.

You can update this server option without stopping and restarting the server by using the **SETOPT** command.

If this value is set to a low number, the activity log might be flooded with messages about the database space being filled, even if there is still space available. If the value is set too high, the database space might be filled before you can add more space to the file system.

### Syntax

►►—FSUSEDThresho1d—*percent*—————►►

### Parameters

*percent*

Specifies the value of used space in the database. You can specify a value from 0 to 100. The default is 90.

### Examples

```
fsusedthreshold 70
```

---

## 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.

The **IDLETIMEOUT** server option is used for non-administrative sessions. See the **ADMINIDLETIMEOUT** option for administrative client sessions.

You can update this server option without stopping and restarting the server by using the **SETOPT** command.

### Syntax

►► IDLETimeout 15  
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
```

---

## KEEPALIVE

The KEEPALIVE option specifies whether the Transmission Control Protocol (TCP) keepalive function is enabled for outbound TCP sockets. The TCP keepalive function sends a transmission from one device to another to check that the link between the two devices is operating.

If you are using node replication, you can use the KEEPALIVE option on the source replication server to enable the TCP keepalive function. The KEEPALIVE option is not required on the target replication server unless you specify bidirectional replication, in which case the target server becomes the source replication server.

### Syntax

►►—KEEPALIVE—Yes  
No—►►

### Parameters

#### *Yes*

Specifies that the TCP keepalive function is enabled for outbound TCP sockets. This value is the default.

If the KEEPALIVE option is enabled, default values are used for the KEEPALIVETIME and KEEPALIVEINTERVAL options.

#### *No*

Specifies that the TCP keepalive function is not enabled for outbound TCP sockets.

If you specify a value of NO, it does not affect current TCP socket connections that originated from outbound connection requests while the KEEPALIVE option was set to YES. The YES value applies to those sockets until the related session ends and the socket is closed.

### Example

Use the SETOPT command to enable the keepalive function without disabling or halting the server:

```
setopt keepalive yes
```

---

## KEEPALIVETIME

The KEEPALIVETIME option specifies how often TCP sends a keepalive transmission when it receives a response. This option applies only if you set the KEEPALIVE option to YES.

### Syntax

►►—KEEPALIVETIME—<sup>300</sup>—seconds—►►

### Parameters

*seconds*

Specifies how often TCP sends keepalive transmissions to verify that an idle connection is still active. The value is specified in seconds.

You can specify a value in the range 1 - 4294967. The default is 300 (5 minutes).

### Example

Set the KEEPALIVETIME option to 120 seconds:

```
keepalivetime 120
```

---

## KEEPALIVEINTERVAL

The KEEPALIVEINTERVAL option specifies how often a keepalive transmission is sent if no response is received. This option applies only if you set the KEEPALIVE option to YES.

### Syntax

►►—KEEPALIVEINTERVAL—<sup>30</sup>seconds—►►

### Parameters

*seconds*

Specifies the length of time, in seconds, between keepalive transmissions when no response is received. The value is specified in seconds.

You can specify a value in the range 1 - 4294967. The default is 30 seconds.

### Example

Set the KEEPALIVEINTERVAL option to 45 seconds:

```
keepaliveinterval 45
```

## 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 that are generated might 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

#### AMENG

Specifies that American English is used as the default language for the server.

#### locale

Specifies the name of the locale that is 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.

Table 466. Server languages for HP-UX

| Language             | LANGUAGE option value |
|----------------------|-----------------------|
| Chinese, Simplified  | zh_CN.hp15CN          |
|                      | zh_CN.utf8            |
| Chinese, Traditional | zh_TW.big5            |
|                      | zh_TW.eucTW           |
|                      | zh_TW.utf8            |
| English              | en_US.iso88591        |
|                      | en_US.utf8            |
| French               | fr_FR.iso88591        |
|                      | fr_FR.utf8            |
| German               | de_DE.iso88591        |
|                      | de_DE.utf8            |
| Italian              | it_IT.iso88591        |
|                      | it_IT.utf8            |
| Japanese             | ja_JP.eucJP           |
|                      | ja_JP.utf8            |
| Korean               | ko_KR.eucKR           |
|                      | ko_KR.utf8            |

Table 466. Server languages for HP-UX (continued)

| Language              | LANGUAGE option value |
|-----------------------|-----------------------|
| Portuguese, Brazilian | pt_PT.iso88591        |
|                       | pt_PT.utf8            |
| Russian               | ru_RU.iso88595        |
|                       | ru_RU.utf8            |
| Spanish               | es_ES.iso88591        |
|                       | es_ES.utf8            |

## Examples

lang ja\_JP.eucJP

---

## LDAPCACHEDURATION

The **LDAPCACHEDURATION** option determines the amount of time that the IBM Tivoli Storage Manager server caches LDAP password authentication information.

After a successful LDAP bind, the value that you enter determines the amount of time that information about the LDAP directory server is kept available. The higher the number, the better the performance of the LDAP directory server. During the cache period, though, changes on the LDAP directory server do not take immediate effect on the node. For example, old passwords might be available for some time, even after they were changed or locked on the LDAP server.

Include the **LDAPCACHEDURATION** option in a **SETOPT** command to have the option take effect immediately.

**Restriction:** The **LDAPCACHEDURATION** option does not apply to storage agents.

### Syntax

▶—LDAPCACHEDURATION—*minutes*—▶

### Parameters

*minutes*

Specifies the maximum amount of time after a successful LDAP bind, that subsequent sessions to the same node or administrator skip secondary LDAP bind operations. Values range from zero to 360 minutes.

### Example: Set the LDAPCACHEDURATION value to 6 hours (maximum)

In the `dsmserv.opt` file, specify the following value:

```
ldapcacheduration 360
```

After a node or administrator authenticates with an external directory server, the LDAP bind is skipped for 360 minutes on all sessions.



---

## LDAPURL

The **LDAPURL** option specifies the location of your LDAP directory server. Set the **LDAPURL** option after you configure the LDAP directory server.

### Restrictions:

- The **LDAPURL** option cannot be used in combination with the **SETOPT** command.
- The **LDAPURL** option does not apply to storage agents.

### Syntax

▶▶—LDAPURL—*ldap\_url\_value*————▶▶

### Parameters

#### *ldap\_url\_value*

Specifies the URL of one LDAP directory server, or the URLs of multiple LDAP directory servers. You can enter multiple values, with each URL value up to 1024 characters. The port number is optional and defaults to 389. Each URL value must contain the following parameters:

- An LDAP directory server name. For example, the format of the server name is `server1.storage.us.ibm.com` and the LDAP port is 389.
- The base distinguished name (Base DN) of the namespace or suffix that the LDAP directory server maintains. Ensure that the DN format conforms to the format that is used on the selected directory server. For example, the format of the Base DN is `ou=tsm,dc=storage,dc=us,dc=ibm,dc=com`.

The value of the LDAPURL option must conform to the following specifications:

- If you specify multiple URLs, each URL must be on a separate line.
- If you specify multiple URLs, each URL must point to a different external directory, and all external directories must contain the same data.
- Each URL must begin with `ldap://`.

**Restriction:** The URL that you designate cannot begin with `ldaps://`.

Tivoli Storage Manager supports LDAP connections that are secured with the standard LDAPv3 StartTLS operation that establishes a secure Transport Layer Security (TLS) exchange on an existing LDAP connection. The LDAP Simple Bind operation that Tivoli Storage Manager uses does not protect the password when it is sent. A secure TLS connection is needed to protect the password.

### Example: Set the LDAP port value for a directory server

In the `dsmserv.opt` file, specify the following port value as 389, for LDAP directory servers, Windows Active Directory servers, and other directory servers:

```
ldapurl ldap://server1.storage.us.ibm.com:389/ou=tsm,dc=storage,dc=us,dc=ibm,dc=com
```

**Tip:** To use the default port, you do not have to specify it.

---

## 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 1190.

### Syntax

▶▶ MAXSessions 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

---

## MIRRORLOGDIRECTORY

The MIRRORLOGDIRECTORY 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

▶—MIRRORlogdirectory—*dir\_name*————▶

### Parameters

*dir\_name*

Specifies a fully qualified directory name for the active log mirror. The maximum number of characters is 175.

### Examples

mirrorlogdirectory /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 number\_of\_files 1000 ◄◄

### 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>4096</sup>—*megabytes*—►►

### Parameters

*megabytes*

Specifies the number of megabytes as an integer from 1 to 32768. The default value is 4096. This option is used 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 minutes<sup>1</sup>►►

### 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

---

## NDMPCONNECTIONTIMEOUT

The NDMPCONNECTIONTIMEOUT server option specifies the time in hours that Tivoli Storage Manager server waits to receive status updates during NDMP restore operations across the LAN. NDMP restore operations of large NAS file systems can have long periods of inactivity. The default is 6 hours.

### Syntax

►►—NDMPCONNECTIONTIMEOUT—6—hours—◄◄

### Parameters

*hours*

The number of hours that the Tivoli Storage Manager server waits to receive status updates during an NDMP restore operation over the LAN. The default value is 6. The minimum is 1 hour. The maximum is 48 hours.

### Example

Specify a timeout of 10 hours before the NDMP connection times out:

```
ndmpconnectiontimeout 10
```



---

## 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—port_number—▶▶
```

The diagram shows the command syntax for the NDMPCONTROLPORT option. It consists of the keyword "NDMPControlport" followed by a parameter "port\_number" enclosed in square brackets. A bracket above the parameter indicates that the default value is 10000. The entire command is shown between two double-headed arrows.

### 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
```

---

## NDMPENABLEKEEPALIVE

The NDMPENABLEKEEPALIVE server option specifies whether the Tivoli Storage Manager server enables Transmission Control Protocol (TCP) keepalive on network data-management protocol (NDMP) control connections to network-attached storage (NAS) devices. The default is NO.

TCP keepalive is implemented within the network support of an operating system. TCP keepalive prevents a long-running, inactive connection from being closed by firewall software that detects and closes inactive connections.

**Restriction:** To prevent errors, do not enable TCP keepalive in certain types of environments. One example is environments that do not have firewalls between the Tivoli Storage Manager server and a NAS device. Another example is environments with firewalls that tolerate long-running, inactive connections. Enabling TCP keepalive in this type of environment can cause an idle connection to be inadvertently closed if the connection partner temporarily fails to respond to TCP keepalive packets.

### Syntax

►►—NDMPENABLEKEEPALIVES—NO—YES—►►

### Parameters

**NO** Disable TCP keepalive on all NDMP control connections. NO is the default.

#### YES

Enable TCP keepalive on all NDMP control connections. The default idle time before the first TCP keepalive packet is sent is 120 minutes.

To change the idle time, issue the following operating system command:

```
ndd -set /dev/tcp tcp_keepalive_interval milliseconds
```

For example, to specify an idle time of fifteen minutes (900,000 milliseconds) before the first TCP keepalive packet is transmitted, issue the following command:

```
ndd -set /dev/tcp tcp_keepalive_interval 900000
```

### Example

Enable TCP keepalive on all NDMP control connections so that inactive NDMP connections are not closed:

```
ndmpenablekeepalive yes
```

---

## 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 that is 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.

### 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 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.

### 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:

```
noretrievevdate
```

---

## 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.

## Syntax

►►—NUMOPENVOLsallowed—*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. The minimum value is 3. The maximum value is 999.

## Examples

Specify that up to 5 volumes in a deduplicated storage pool can be open at one time.

```
numopenvolallowed 5
```

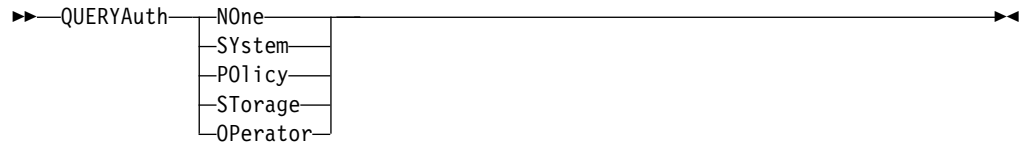


---

## 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.

#### **P0l1cy**

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.

#### **0Perator**

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—<sup>30</sup>*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
```

---

## REORGBEGINTIME

The REORGBEGINTIME option specifies the earliest time that the Tivoli Storage Manager server can start a table or index reorganization.

Schedule server-initiated reorganizations to start during periods when server activity is low. Use this option together with the REORGDURATION option. The REORGDURATION specifies an interval during which reorganization can start.

### Syntax

►►—REORGBEGINTime—*hh:mm*—►►

### Parameters

*hh:mm*

Specifies the time that the server can start a reorganization: The default start time 6:00 a.m. Use a 24-hour format to specify the time.

| Time      | Description            | Values                    |
|-----------|------------------------|---------------------------|
| <i>hh</i> | The hour of the day    | Specify a number 00 - 23. |
| <i>mm</i> | The minute of the hour | Specify a number 00 - 59. |

### Examples

Specify 6:00 a.m. as the earliest time that a reorganization can start.  
reorgbegintime 06:00

Specify 8:30 p.m. as the earliest time that a reorganization can start.  
reorgbegintime 20:30

Specify noon as the earliest time that a reorganization can start.  
reorgbegintime 12:00

Specify 3:30 p.m. as the earliest time that a reorganization can start.  
reorgbegintime 15:30

Specify midnight as the earliest time that a reorganization can start.  
reorgbegintime 00:00

---

## REORGURATION

The REORGURATION option specifies an interval during which server-initiated table or index reorganization can start.

Schedule server-initiated reorganizations to start during periods when server activity is low. Use this option together with the REORGBEGINTIME option. The REORGBEGINTIME option specifies the earliest time that the server can start a reorganization.

### Syntax

►►—REORGURATION—*nn*—————►►

### Parameters

*nn* Specifies the number of hours during which a reorganization can start. The minimum value is 1, the maximum value is 24. The default value is 24.

### Example

Specify an interval of four hours during which a reorganization can start.

```
reorgduration 4
```

---

## 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
```

---

## REPLBATCHSIZE

The REPLBATCHSIZE option specifies the number of client files that are to be replicated in a batch, within the same server transaction. This option affects only the node replication processes and works with the REPLSIZETHRESH option to improve node replication processing.

The REPLBATCHSIZE option limits the number of files in a transaction and the REPLSIZETHRESH option limits the number of bytes in a transaction. The transaction ends when either the REPLBATCHSIZE threshold or the REPLSIZETHRESH threshold is reached.

### Syntax

►►—REPLBatchsize—<sup>4096</sup>*number\_of\_files*—►►

### Parameters

*number\_of\_files*

Specifies a number of files between 1 - 32768. The default is 4096.

### Examples

replbatchsize 25000

---

## REPLSIZETHRESH

The REPLSIZETHRESH option specifies, in megabytes, a threshold for the amount of data replicated, within the same server transaction.

The amount of data is based on the non-deduplicated size of the file, which is the original size of the file. The amount of data that is replicated is controlled by the threshold. When the amount of data exceeds the threshold, the server ends the transaction and no more files are added to the current batch. A new transaction is started after the current batch is replicated. This option is used with the REPLBATCHSIZE option.

For example, suppose that a file is 10 MB and is stored in a data-deduplication-enabled storage pool and only 2 MB of the file is transferred during replication. The amount of data replicated includes the 10 MB size of the file, and excludes the 2 MB transferred. When the amount of data replicated exceeds the value specified for the REPLSIZETHRESH threshold, the transaction ends.

### Syntax

►►—REPLSizethresh—4096—*megabytes*—►►

### Parameters

*megabytes*

Specifies the number of megabytes as an integer from 1 - 32768. The default value is 4096.

### Examples

replsizethresh 2000



---

## 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
- GENERATE 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—60  
*minutes*—►►

### 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 1190.

### Syntax

►►—RESTOREINTERVAL minutes 1440—►►

### 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
```

---

## 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
```

---

## SERVEDEDUPTXNLIMIT

The SERVEDEDUPTXNLIMIT option specifies the maximum size of objects that can be deduplicated on the server.

When you use duplicate-identification processes (the **IDENTIFY DUPLICATES** command) for large objects, intensive database activity can result from long-running transactions that are required to update the database. High levels of database activity can produce following symptoms:

- Reduced throughput for client backup and archive operations
- Resource contention resulting from concurrent server operations
- Excessive recovery log activity

The extent to which these symptoms occur depends on the number and size of objects being processed, the intensity and type of concurrent operations taking place on the Tivoli Storage Manager server, and the Tivoli Storage Manager server configuration.

With the SERVEDEDUPTXNLIMIT server option, you can specify a maximum size, in gigabytes, for objects that can be deduplicated on the server. If an object or set of objects in a single transaction exceeds the limit specified by SERVEDEDUPTXNLIMIT, the objects are not deduplicated by the server. You can specify a value 32 - 102400 GB. The default value is 5120 GB.

Increasing the value of this option causes the Tivoli Storage Manager server to search for objects previously deferred whose size falls below the new transaction limit.

**Remember:** The search for objects previously deferred can take time. Use care when increasing the value of SERVEDEDUPTXNLIMIT. Reducing the value of this option does not cause Tivoli Storage Manager to search for deferred objects.

The appropriate value for this option depends on the Tivoli Storage Manager server configuration and concurrent server activity. You can specify a high value for this option if you minimize resource contention. To minimize resource contention, perform operations, such as backup, archive, duplicate identification, and reclamation, at different times.

To update this server option without stopping and restarting the server, use the **SETOPT** command.

### Syntax

►►—SERVEDEDUPTXNLIMIT—<sup>5120</sup>—*gigabytes*—►►

### Parameters

*gigabytes*

Specifies the maximum size, in gigabytes, of objects that can be duplicated on the server. You can specify a value 32 - 102400. The default value is 5120.

## Examples

Disable server-side deduplication for all objects over 120 GB:

```
serverdeduptxnlimit 120
```

---

## 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
```

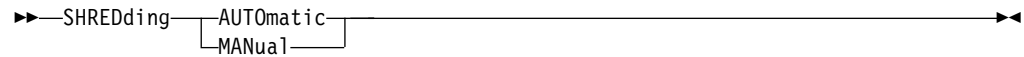


---

## 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

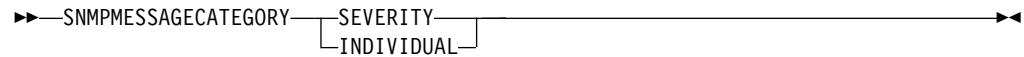
```
snmpheartbeatinterval 20
```

---

## 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

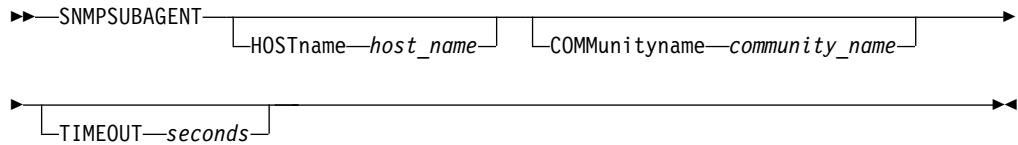
```
snmpmessagecategory individual
```

---

## 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
```

## SSLDISABLELEGACYTLS

The SSLDISABLELEGACYTLS option specifies whether to use the Transport Layer Security (TLS) 1.2 or later protocol for Secure Sockets Layer (SSL) sessions. The server rejects connection attempts that use levels earlier than TLS 1.2.

### Syntax



### Parameters

#### Yes

Specifies that the server uses the TLS 1.2 or later protocol for SSL sessions.

The SSLDISABLELEGACYTLS option overrides the SSLTLS12=NO option and enforces the rejection of SSL connection attempts that use levels earlier than TLS 1.2.

**Requirements:** Before you use TLS 1.2, ensure that the following settings are correct:

- For the server and storage agent, if you use self-signed certificates, you must set the default label in the key database to “TSM Server SelfSigned SHA Key”.
- For backup-archive clients, if you use self-signed certificates, you must import the cert256.arm file.

**No** Specifies that the server allows TLS 1.1 and earlier protocol for SSL sessions. Specify the SSLTLS12=YES option to allow the server to use TLS 1.2 in addition to earlier protocols.

If you specify the SSLTLS12=YES option and do not specify the SSLDISABLELEGACYTLS option, TLS 1.2 might be used.

Table 467. TLS versions used by the server for the SSLTLS12 and SSLDISABLELEGACYTLS options

| SSLTLS12 | SSLDISABLELEGACYTLS | TLS version that is used by the server |
|----------|---------------------|----------------------------------------|
| No*      | No*                 | ≤ TLS 1.1                              |
| No*      | Yes                 | ≥ TLS 1.2                              |
| Yes      | No*                 | ≤ TLS 1.2                              |
| Yes      | Yes                 | ≥ TLS 1.2                              |

**Notes:**

1. An asterisk (\*) indicates the default for an option.
2. ≤ A less than or equal symbol indicates the highest TLS version that is used by the server.
3. ≥ A greater than or equal symbol indicates the lowest TLS version that is used by the server.

## Examples

Specify that the server uses the TLS 1.2 or later protocol for SSL sessions:

```
ssldisablelegacytls yes
```

Specify that the server allows TLS 1.1 and earlier protocol for SSL sessions:

```
ssldisablelegacytls no
```

---

## SSLFIPSMODE

The SSLFIPSMODE option specifies whether the Federal Information Processing Standards (FIPS) mode is in effect for Secure Sockets Layer (SSL). The default is NO.

Because SSLv3 is not supported by FIPS mode, when you are using SSL with Version 6.1 or V5.5 clients, you must turn off FIPS mode.

### Syntax



### Parameters

**No** Specifies that SSL FIPS mode is not active on the server. This setting is required when Backup-Archive Client versions previous to IBM Tivoli Storage Manager 6.3 are to connect to the server with SSL.

**Yes**

A value of YES indicates that SSL FIPS mode is active on the server. This setting restricts SSL session negotiation to use FIPS-approved cipher suites. Specifying YES is suggested when SSL communication is activated and all Backup-Archive Clients are at V6.3 or later.

**Remember:** Set server option SSLTLS12 to NO if your environment includes Backup-Archive Clients that use server self-signed certificates that were generated by a server whose version was before V6.3.

To disable SSL FIPS mode on the server:

```
SSLFIPSMODE no
```



---

## SSLHIDELEGACYTLS

The SSLHIDELEGACYTLS option specifies whether to restrict the availability of Transport Layer Security (TLS) 1.1 and earlier protocols for Secure Sockets Layer (SSL) ports on which the server or storage agent listens.

You can use this option to ensure that failures to use TLS 1.1 and earlier protocols for connections are not logged as errors. By using the option, you can also determine whether any clients are not configured for using TLS 1.2.

This option is useful in scenarios such as the following one:

1. You want to restrict secure communications to use TLS 1.2. In this way, you prevent the use of previous TLS protocol levels, which are less secure. To restrict communications to TLS 1.2, you specify a value of YES for the SSLDISABLELEGACYTLS option.
2. To ensure that failures to use earlier protocols are not logged as errors, you specify YES for the SSLHIDELEGACYTLS option.

**Restriction:** The SSLHIDELEGACYTLS option applies only if you specify YES for the SSLDISABLELEGACYTLS option.

### Syntax



### Parameters

#### Yes

Servers and storage agents preclude the use of TLS 1.1 or earlier protocols for SSL ports. Failed attempts to use TLS 1.1 or earlier protocols do not generate error messages in the activity log.

**No** Servers and storage agents detect TLS 1.1 or earlier protocols for SSL ports, even if sessions that use those protocols are disabled. This is the default setting.

### Example

Specify that servers and storage agents do not detect TLS 1.1 or earlier protocols for SSL ports:

```
ssldisablelegacytls yes
sslhidelegacytls yes
```

---

## SSLINITTIMEOUT

The SSLINITTIMEOUT option specifies the time, in minutes, that the server waits for a Secure Sockets Layer (SSL) session to complete initialization before the server cancels the session.

When you specify this option, an SSL session is canceled if a client, server, or storage agent is not configured for SSL and tries to start an SSL session. Similarly, an SSL session is canceled if a client SSL session and a server are not configured with the same Transport Layer Security (TLS) version. In these situations, the SSL session might fail to completely initialize. The server cancels the session when the specified timeout is reached.

### Syntax

►►—SSLINITTIMEout—<sup>2</sup>minutes—►►

### Parameters

*minutes*

Specifies the maximum number of minutes that a server waits for an SSL session to complete initialization. The default value is 2 minutes. The minimum value is 1 minute.

### Example

```
sslinittimeout 1
```

---

## SSLTCPADMINPORT

The SSLTCPADMINPORT option specifies the port address on which the server TCP/IP communication driver waits for requests for SSL-enabled sessions. The sessions are for the command-line administrative client.

First-time use of the SSLTCPADMINPORT or SSLTCPPOINT option triggers the creation of a key database file (`cert.kdb`) at server startup. The key database file is created in the server instance directory and is initialized with a self-signed certificate.

The following types of sessions do not use SSL (Secure Sockets Layer):

- NDMP (Network Data Management Protocol)
- ACSLS (Automated Cartridge System Library Software)
- SNMP (Simple Network Management Protocol) subagent
- Database restore operations

If the ADMINONCLIENTPORT option is set to NO, SSL-enabled sessions for the administrative client require SSLTCPADMINPORT with a port number other than one specified by the SSLTCPPOINT option. The SSLTCPADMINPORT 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 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
```

---

## SSLTCPPORT

The SSLTCPPORT option specifies the SSL (Secure Sockets Layer) port number. The server TCP/IP communication driver waits for requests on this port for SSL-enabled sessions from the client.

First-time use of the SSLTCPADMINPORT or SSLTCPPORT options triggers the creation of a key database file (`cert.kdb`) at server startup. The key database file is created in the server instance directory and is initialized with a self-signed certificate.

The following types of sessions do not use SSL:

- NDMP (Network Data Management Protocol)
- ACSLS (Automated Cartridge System Library Software)
- SNMP (Simple Network Management Protocol) subagent
- Database restore operations

If the ADMINONCLIENTPORT option is set to NO, SSL-enabled sessions for the administrative client require SSLTCPADMINPORT with a port number different from one specified by the SSLTCPPORT option. The SSLTCPPORT 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

▶—SSLTCPPort—*port\_number*—▶

### Parameters

*port\_number*

Specifies the port number of the server. Valid values are 1024 - 32767. There is no default.

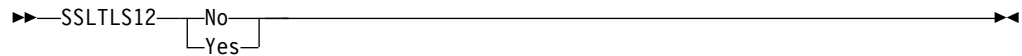
### Examples

```
ssltcpport 1542
```

## SSLTLS12

The SSLTLS12 option specifies whether to use the Transport Layer Security (TLS) 1.2 or later protocol for Secure Sockets Layer (SSL) sessions.

### Syntax



### Parameters

#### Yes

Specifies that the server might use the TLS 1.2 or later protocol for SSL sessions. If you specify the SSLTLS12=YES option and do not specify the SSLDISABLELEGACYTLS option, TLS 1.2 might be used.

**No** Specifies that the server uses TLS 1.1 or earlier protocol for SSL sessions.

The SSLDISABLELEGACYTLS option overrides the SSLTLS12=NO option and enforces the rejection of SSL connection attempts that use levels earlier than TLS 1.2.

*Table 468. TLS versions used by the server for the SSLTLS12 and SSLDISABLELEGACYTLS options*

| SSLTLS12 | SSLDISABLELEGACYTLS | TLS version that is used by the server |
|----------|---------------------|----------------------------------------|
| No*      | No*                 | ≤ TLS 1.1                              |
| No*      | Yes                 | ≥ TLS 1.2                              |
| Yes      | No*                 | ≤ TLS 1.2                              |
| Yes      | Yes                 | ≥ TLS 1.2                              |

**Notes:**

1. An asterisk (\*) indicates the default for an option.
2. ≤ A less than or equal symbol indicates the highest TLS version that is used by the server.
3. ≥ A greater than or equal symbol indicates the lowest TLS version that is used by the server.

### Example

Set TLS 1.2 as the protocol for SSL sessions:

```
SSLTLS12 yes
```

---

## 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 TCPSPORT 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 TCPSPORT 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 TCPSPORT 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 TCPSPORT.

### 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 value range is from 1 to 64. The default is 16.

### 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

►—TCPNodeLAY—Yes  
                  No

### 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
```



---

## TCPPOINT

The TCPPOINT 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 TCPPOINT 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.

### Syntax

▶▶—TCPPOINT—*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.

### 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 63
```

---

## 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

►►—`TECBeginEventLogging` Yes  
No—————►

### 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.

### 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 1190.

### 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 1190.

### 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 989 and “UPDATE NODE (Update node attributes)” on page 1320.

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 4096

---

## 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

►►—UNIQUETDPtecevents—

|     |
|-----|
| Yes |
| No  |

—————►

### 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

►►—UNIQUETEEvents  Yes  No ►

### 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

```
uniquetecevents yes
```

---

## USEREXIT

The USEREXIT option specifies a user-defined exit that will be given control to manage an event.

### Syntax

►►—USEREXIT—┌ Yes ───┐  
└ No ───┘ *module\_name*—►►

### 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 469. Server utilities*

| Utility                                                                                   | Description                                                        |
|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| "DSMSERV (Start the server)" on page 1586                                                 | Starts the server.                                                 |
| "Server startup script: rc.dsmserv" on page 1588                                          | Automatically starts a server instance.                            |
| "DSMSERV DISPLAY DBSPACE (Display information about database storage space)" on page 1589 | Displays information about storage space defined for the database. |
| "DSMSERV DISPLAY LOG (Display recovery log information)" on page 1590                     | Displays information about recovery log storage space.             |
| "DSMSERV FORMAT (Format the database and log)" on page 1594                               | Initializes the database and recovery log.                         |
| "DSMSERV INSERTDB (Move a server database into an empty database)" on page 1597           | Inserts a server database into a new Version 6 database.           |
| "DSMSERV LOADFORMAT (Format a database)" on page 1600                                     | Formats an empty database.                                         |
| "DSMSERV REMOVEDB (Remove a database)" on page 1602                                       | Removes a Tivoli Storage Manager database.                         |
| "DSMSERV RESTORE DB (Restore the database)" on page 1604                                  | Restores a Tivoli Storage Manager database.                        |

---

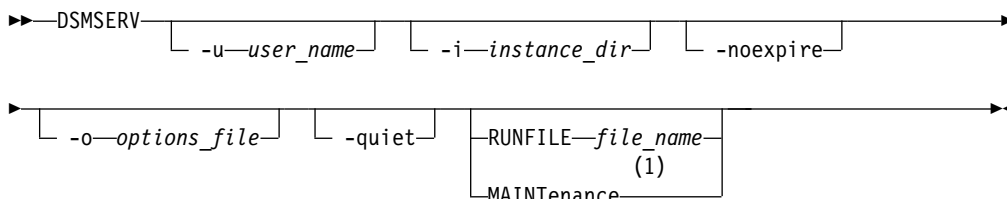
## DSMSERV (Start the server)

Use this utility to start the Tivoli Storage Manager server.

### Restrictions:

- Do not enter more than 1022 characters in the DSMSERV console command-line interface. Text that exceeds 1022 characters is truncated.

### Syntax



### Notes:

- 1 This parameter applies only to AIX, Linux, Solaris, and Windows servers.

### Parameters

#### **-u** *user\_name*

Specifies a user name to switch to before you start the server. To start the server from the root user ID, you must specify the **-u** parameter and follow the instructions about starting the server from the root user ID ([http://www.ibm.com/support/knowledgecenter/SSGSG7\\_7.1.1/com.ibm.itsm.srv.doc/t\\_srv\\_mng\\_startsrv\\_root.html](http://www.ibm.com/support/knowledgecenter/SSGSG7_7.1.1/com.ibm.itsm.srv.doc/t_srv_mng_startsrv_root.html)).

#### **-i** *instance\_dir*

Specifies an instance directory to use. The instance directory becomes the current working directory of the server.

#### **-noexpire**

Specifies that the server does not remove expired files from the server database. The files are not deleted from server storage when you start the server.

#### **-o** *options\_file*

Specifies an options file to use.

#### **-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 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. Issue the following command on one line:

```
/opt/tivoli/tsm/server/bin/dsmserv
```



### **Example: Load the sample script**

Load the sample script file that is provided with the server.

```
dmserv runfile scripts.smp
```

---

## Server startup script: rc.dsmserv

You can use the **rc.dsmserv** script 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 instance user ID for which the environment is set up. The server will run under this user ID.

**-U** *user\_name*

Specifies the 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, which becomes the 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  
-noexpire -quiet DISPLAY LOG
```

### 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  
Archive Log Directory: /archivelog  
Mirror Log Directory: /mirrorlog  
Archive Failover Log Directory: /archfailoverlog
```

### 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.

**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**

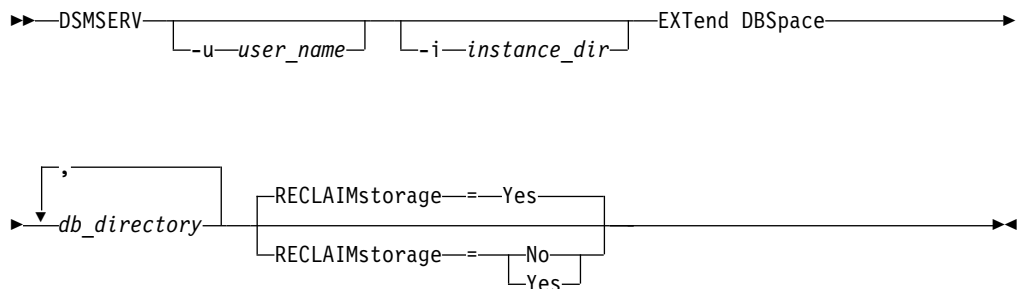
Specifies the location in which the server saves archive logs if the logs cannot be archived to the archive log destination.

## DSMSERV EXTEND DBSPACE (Increase space for the database)

Use this utility to increase space for the database by adding directories for the database to use. This utility performs the same function as the **EXTEND DBSPACE** command, but you can use it when the server is not running.

**Restriction:** Redistribution of data and reclaiming of space as part of an operation to extend database space only works with DB2 Version 9.7 or later table spaces, which are created when you format a new Tivoli Storage Manager Version 6.2 or later server. If you upgraded or restored your Tivoli Storage Manager server from V6.1, you cannot redistribute data or reclaim space. You must use the **DSMSERV EXTEND DBSPACE** utility with **RECLAIMSTORAGE=NO**.

### Syntax



### Parameters

**-u** *user\_name*

Specifies a user name to switch to before you initialize the server.

**-i** *instance\_dir*

Specifies an instance directory to use. This becomes the current working directory of the server.

**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 and cannot exceed 175 characters in length. Enclose the name in quotation marks if it contains embedded blanks, an equal sign, or other special characters. If you are specifying a list of directories for database storage, the maximum length of the list can be 1400 characters.

**Tip:** Specify directories that are the same size as existing directories to ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching and distribution of the database.

### RECLAIMstorage

Specifies whether data is redistributed across newly created database directories and space is reclaimed from the old storage paths when you add space to the database. This parameter is optional. The default value is Yes.

**Yes**

Specifies that data is redistributed so that new directories are available for immediate use.

**Important:** The redistribution process uses considerable system resources so ensure that you plan ahead. Also, the server might be offline for a while, until the process is completed.

**No** Specifies that data is not redistributed across database directories and storage space is not reclaimed.

### **Example: Increase space for the database**

Add a directory named `stg1` in the `tsm_db` directory for the database storage space and then redistribute data and reclaim space by issuing the following command:

```
dsmserv extend dbspace /tsm_db/stg1
```

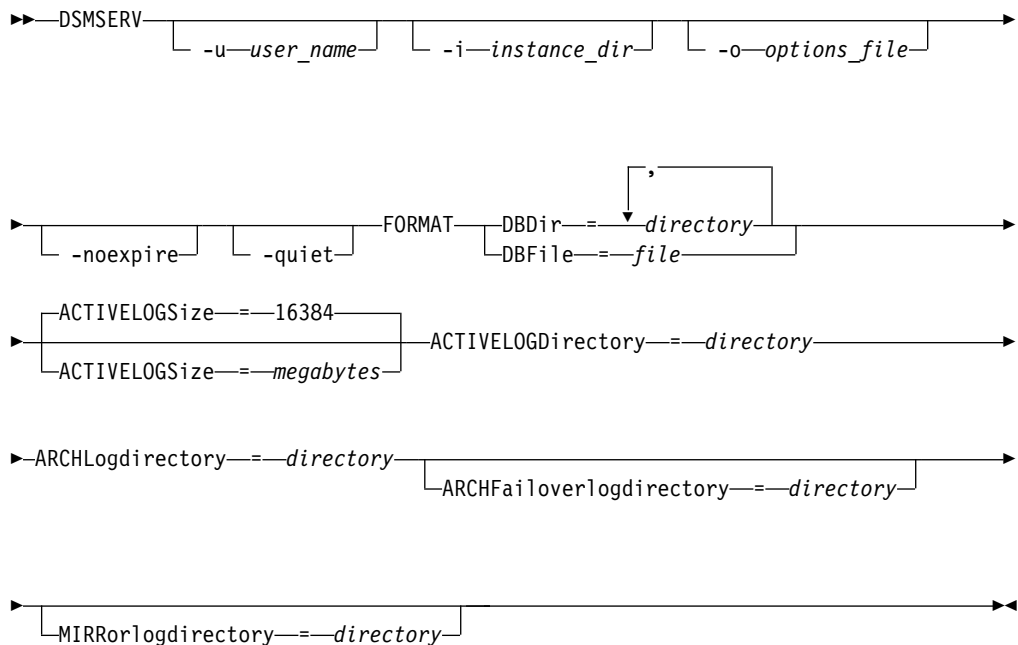
## DSMSERV FORMAT (Format the database and log)

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.

When a server is initially created by using the **DSMSERV FORMAT** utility or the configuration wizard, a server database and recovery log are created. In addition, files are created to hold database information that is used by the database manager.

### 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 directory 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.

**Tip:** If you specify multiple directories, ensure that the underlying file systems are of equal size to ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching and distribution of the database.

**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 file in megabytes. This parameter is optional. The minimum value is 2048 MB (2 GB); the maximum is 524,288 MB (512 GB). If an odd number is specified, the value is rounded up to the next even number. The default is 16384 MB.

The size of an active log file that is used by Tivoli Storage Manager is based on the value of the **ACTIVELOGSIZE** option. Guidelines for space requirements are in the following table:

*Table 470. How to estimate volume and file space requirements*

| <b>ACTIVELOGSize option value</b> | <b>Reserve this much free space in the active log directory, in addition to the ACTIVELOGSize space</b> |
|-----------------------------------|---------------------------------------------------------------------------------------------------------|
| 2 GB - 128 GB                     | 5120 MB                                                                                                 |
| 129 GB - 256 GB                   | 10240 MB                                                                                                |
| 257 GB - 512 GB                   | 20480 MB                                                                                                |

**ACTIVELOGDirectory (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 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.

**ARCHLogdirectory (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.

**ARCHFailoverlogdirectory**

Specifies the directory to be used as an alternative storage location if the **ARCHLOGDIRECTORY** directory is full. This parameter is optional. The maximum number of characters is 175.

**MIRRORlogdirectory**

Specifies the directory in which the server mirrors the active log (those files in

the ACTIVELOGDIRECTORY 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 activelogsiz=8192  
activelogdirectory=/active log archlogdirectory=/archlog  
archfailoverlogdirectory=/archfaillog mirrorlogdirectory=/mirrorlog
```

## DSMSERV INSERTDB (Move a server database into an empty database)

Use the **DSMSERV INSERTDB** utility to move a server database into a new 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 two servers. The database can also be inserted from media that contains the extracted database.

Before you use the **DSMSERV INSERTDB** utility, complete the planning and preparation tasks, such as backing up the database and saving configuration information. Ensure that you meet all requirements before you move the server database.

### Requirements for insertion by 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.
- If the manifest file does not contain device configuration information, or if you are specifying the **CONFIGINFO=DEVCONFIG** parameter, both of the following statements must be true:
  - 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 V7 server. Also, the permissions must be set to grant access to the media for the user ID that owns the V7 server instance.

### Syntax

```
DSMSERV [-u user_name] [-i instance_dir] [-o options_file]
```

```
[-noexpire] [-quiet] INSERTDB [A: Insert from media] [B: Insert over a network]
```

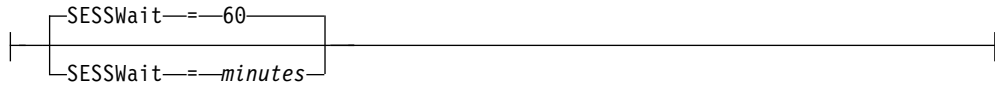
```
PREview == No
PREview == Yes
           No
```

#### A: Insert from media:

```
[-DEVclass == device_class_name] [-CONFIGinfo == MANifest]
[-CONFIGinfo == MANifest]
[-CONFIGinfo == DEVconfig]
```

```
MANifest == file_name
```

## 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 directory 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 either the manifest file or the device configuration file.

This parameter is optional and is used only when the database that you want to insert into the empty V7 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 V7 database was extracted to media.

### CONFiginfo

Specifies the source of the device configuration information that is used by the **DSMSERV INSERTDB** operation. The default value for this parameter is **MANIFEST**. Possible values are as follows:

#### MANifest

Specifies that device configuration information is read from the manifest file. If the manifest file does not have device configuration information, the device configuration file is used instead.

#### DEVConfig

Specifies that device configuration information is read from the device configuration file.

**SESSWait**

Specifies the number of minutes that the V7 server waits to be contacted by the original server. The default value is 60 minutes.

Use this parameter only if the data that is inserted into the empty V7 database is transmitted from the source server with a network connection.

**PREview**

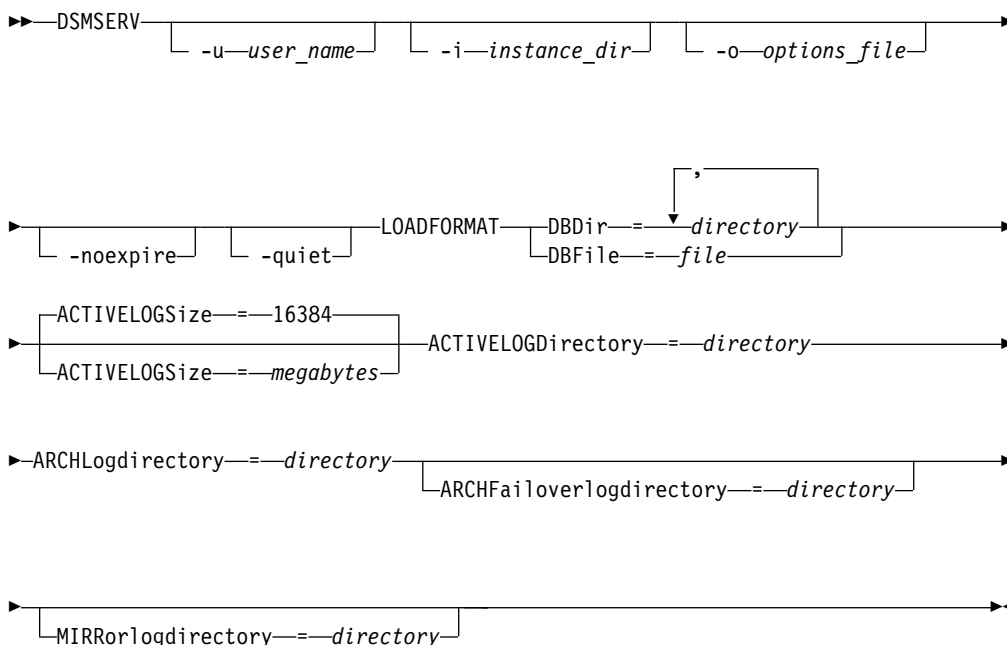
Specifies whether to preview the insertion operation. This parameter is optional. The default value is **NO**.

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.

### 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 directory 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.

**Tip:** If you specify multiple directories, ensure that the underlying file systems are of equal size to ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching and distribution of the database.

**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 file in megabytes. This parameter is optional. The minimum value is 2048 MB (2 GB); the maximum is 524,288 MB (512 GB). If an odd number is specified, the value is rounded up to the next even number. The default is 16384 MB.

The size of an active log file that is used by Tivoli Storage Manager is based on the value of the ACTIVELOGSIZE option. Guidelines for space requirements are in the following table:

*Table 471. How to estimate volume and file space requirements*

| <b>ACTIVELOGSize option value</b> | <b>Reserve this much free space in the active log directory, in addition to the ACTIVELOGSize space</b> |
|-----------------------------------|---------------------------------------------------------------------------------------------------------|
| 2 GB - 128 GB                     | 5120 MB                                                                                                 |
| 129 GB - 256 GB                   | 10240 MB                                                                                                |
| 257 GB - 512 GB                   | 20480 MB                                                                                                |

**ACTIVELOGDirectory (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 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.

**ARCHLogdirectory (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.

**ARCHFailoverlogdirectory**

Specifies the directory to be used as an alternative storage location if the ARCHLOGDIRECTORY directory is full. This parameter is optional. The maximum number of characters is 175.

**MIRRORlogdirectory**

Specifies the directory in which the server mirrors the active log (those files in the ACTIVELOGDIRECTORY 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**

```
dmserv loadformat dbdir=/tsmdb001 activelogsiz=8192
activelogdirectory=/activelog archlogdirectory=/archlog
archfailoverlogdirectory=/archfaillog mirrorlogdirectory=/mirrorlog
```

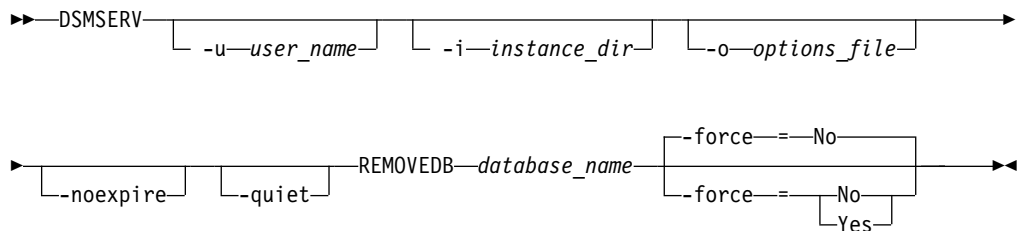
## DSMSERV REMOVEDB (Remove a database)

Use the DSMSERV REMOVEDB utility to remove a Tivoli Storage Manager server database.

When you run this utility, you delete the server database, active log files, and active log mirror files. However, the archive log files and archive log failover log files are deleted only after you start a point-in-time database restore.

You must halt the Tivoli Storage Manager server before you issue this command.

### 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.

*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.

**-force**

Specifies whether the database is removed when there are open connections. The default is No. This parameter is optional. The values are as follows:

**Yes**

Specifies that the database is removed regardless of open connections

**No**

Specifies that the database is removed only when all connections are closed.

### Example: Remove a database

Remove the Tivoli Storage Manager server database TSMDB1 and all of its references.



```
dsmserv removedb TSMDB1
```

### **Example: Remove a database with force parameter**

Remove the Tivoli Storage Manager server database TSMDB1 and all of its references, even if it has open connections:

```
dsmserv removedb TSMDB1 force=yes
```

---

## DSMSERV RESTORE DB (Restore the database)

Use this utility to restore a database by using a database backup.

**Restrictions:** You cannot restore a server database if the release level of the server database backup is different from the release level of the server that is being restored. For example, an error occurs when you restore a Version 6.3 database and you are using a Version 7.1 Tivoli Storage Manager server.

You cannot restore a server database from a different Tivoli Storage Manager server if both of the following conditions are present:

- The database backup is stored on virtual volumes
- The connection to the Tivoli Storage Manager server is protected by SSL (Secure Sockets Layer)

Use this utility for the following tasks:

- “DSMSERV RESTORE DB (Restore a database to its most current state)” on page 1605
- “DSMSERV RESTORE DB (Restore a database to a point-in-time)” on page 1608

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.

**Restrictions:** You cannot restore a server database if the release level of the server database backup is different from the release level of the server that is being restored. For example, an error occurs when you restore a Version 6.3 database and you are using a Version 7.1 Tivoli Storage Manager server.

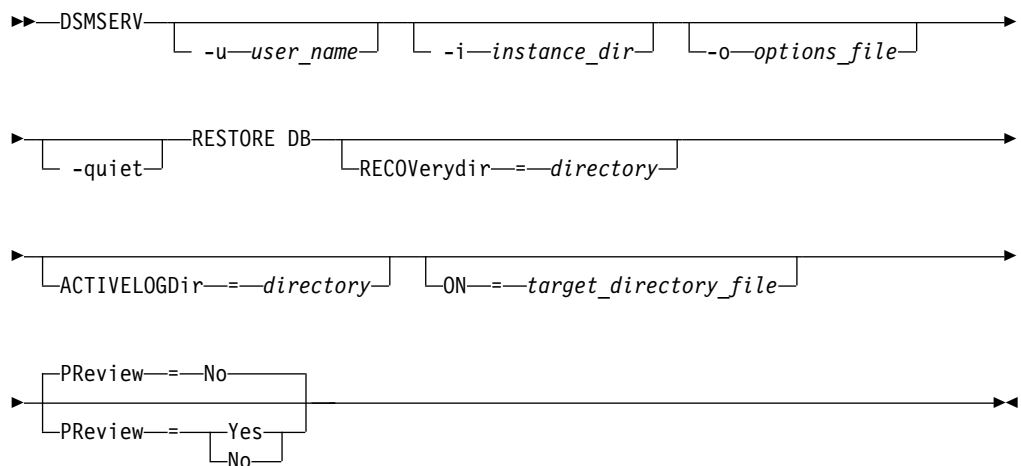
You cannot restore a server database from a different Tivoli Storage Manager server if both of the following conditions are present:

- The database backup is stored on virtual volumes
- The connection to the Tivoli Storage Manager server is protected by SSL (Secure Sockets Layer)

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



### 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 instance directory becomes the current working directory of the server.

**-o options\_file**  
Specifies an options file to use.

**-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:

- ARCHFAILOVERLOGDIRECTORY, if specified
- ARCHLOGDIRECTORY, if ARCHFAILOVERLOGDIRECTORY is not specified

#### **ACTIVELOGdir**

Specifies a directory in which to store the log files that are used to track the active database operations. This directory must 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 that lists the directories to which the database is 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.

**Tip:** If you specify multiple directories, ensure that the underlying file systems are of equal size to ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching and distribution of the database.

#### **PReview**

Specifies that the volume history files be examined and that the database backup volumes from the volume history file be evaluated.

1. Which set of database backup volumes best meets the most current criteria that are specified for restore processing? The volume history information provides details about the backup series ID, the operation ID (full, incremental 1, incremental 2, and so on), the date of the database backup, and the device class. This information and the parameters that are specified in the **DSMSERV RESTORE DB** command determine what to use to perform the restore. The volume history file is examined to find the most recent database backup and then to restore the data by using that backup.
2. Is self-describing data available for the selected set of database backup volumes? Cross-check the volume history information for this backup series. The reconciliation reports what the self-describing data contains compared to what was learned from the volume history entries. The cross-check involves mounting one or more of the volumes that are indicated by the volume history. Then, using the self-describing data that was included in the database backup volumes, that information is reconciled against what is in the volume history for the database backup. If

the information from the volume history file is inconsistent with the self-describing data, then messages are issued to identify the problem. For example, not all values are specified and available, and no self-describing data is found.

If the volume history information is consistent with self-describing data from the database backup, a message is issued indicating that the database backup can be used for restore processing.

If the volume history information is inconsistent with the self-describing data from the database backup or if the self-describing data for the backup cannot be found, error messages are issued indicating what was checked and what was missing.

If the `PREVIEW` parameter is not specified or if it is set to `NO`, and if the volume history and self-describing data from the database backup are consistent, then the restore proceeds.

If the `PREVIEW` parameter is not specified or if it is set to `NO`, and the reconciliation and validation fail, the database restore is not performed. Make extra volumes available and referred to from the volume history file, or remove the incomplete backup series or operation so that the Tivoli Storage Manager server selects a different preferred series or operation and continues processing.

If the `PREVIEW` parameter is set to `YES`, the process performs only the evaluation of the volume history file and the reconciliation and validation against the selected database backup.

### **Example: Restore the database to its most current state**

Restore the database to its most current state by using the already configured active log directory.

```
dsmserv restore db
```

### **Example: Restore the server master key without restoring the database**

Restore the server master key without restoring the database by issuing the following command:

```
dsmserv restore db restorekeys=only
```

## DSMSERV RESTORE DB (Restore a database to a point-in-time)

Use this command to restore a database to a point in time. A volume history file and a device configuration file must be available.

**Restrictions:** You cannot restore a server database if the release level of the server database backup is different from the release level of the server that is being restored. For example, an error occurs when you restore a Version 6.3 database and you are using a Version 7.1 Tivoli Storage Manager server.

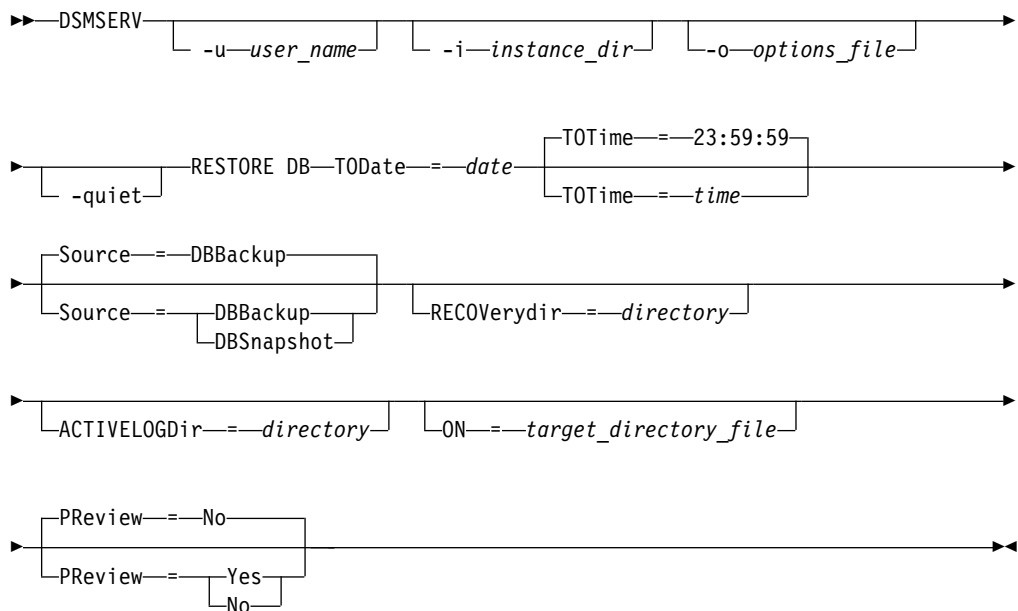
You cannot restore a server database from a different Tivoli Storage Manager server if both of the following conditions are present:

- The database backup is stored on virtual volumes
- The connection to the Tivoli Storage Manager server is protected by SSL (Secure Sockets Layer)

You can use full and incremental database backups, or snapshot database backups can be used to restore a database to a point in time.

**Tip:** When you restore a V6 or later Tivoli Storage Manager server database to a specific point in time, the preferred method is to issue the **DSMSERV REMOVE DB** command before you issue the **DSMSERV RESTORE DB** command. This ensures that the system is in a clean state. The system drops and uncatalogs the database in the background. When you restore data to a specific point in time, all the required logs and the database image are retrieved from the backup media.

### Syntax



### Parameters

**-u** *user\_name*  
Specifies a user name to switch to before you initialize 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.

**-quiet**

Specifies that messages to the console are suppressed.

**TODate (Required)**

Specifies the date to which to restore the database. The following values are possible:

*MM/DD/YYYY*

Specifies that you want to restore a database by using the last backup series that was created before this specified date.

**TODAY**

Specifies that you want to restore a database by using the most recent backup series that was created before today.

**TODAY-*numdays* or -*numdays***

Specifies that you want to restore a database by 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 by using the last backup series that is 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 by using a backup series that is created on or before the current time on the date that is specified on the TODATE parameter.

For example, if you issue the DSMSEV RESTORE DB utility at 9:00 with TOTIME=NOW, the database is restored by using the last backup series that is created on or before 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 by using a backup series that is 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 DSMSEV RESTORE DB utility at 9:00 with TOTIME=NOW-3:30 or TOTIME+-3:30, the database is restored by using the last backup series that is created on or before 5:30 on the date that is specified on the TODATE parameter.

**Source**

Specifies whether the database is restored by using either database full and incremental backup volumes or snapshot database volumes. This parameter is optional. The default value is DBBackUp. The following values are possible:

**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 DSMSEV FORMAT or DSMSEV LOADFORMAT utility:

- ARCHFAILOVERLOGDIRECTORY, if specified
- ARCHLOGDIRECTORY, if ARCHFAILOVERLOGDIRECTORY 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 was already configured.

- On** Specifies a file that lists the directories to which the database is 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.

**Tip:** If you specify multiple directories, ensure that the underlying file systems are of equal size to ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching and distribution of the database.

**PReview**

Specifies that the volume history files be examined and that the database backup volumes from the volume history file be evaluated.

1. Which set of database backup volumes best meets the point-in-time criteria that are specified for restore processing? The volume history information provides details about the backup series ID, the operation ID (full, incremental 1, incremental 2, and so on), the date of the database backup, and the device class. This information and the parameters that are specified



in the **DSMSERV RESTORE DB** command determine what to use to perform the restore. The volume history file is examined to find the best database backup that meets the specified point-in-time criteria and then perform the restore by using that backup.

2. Is self-describing data available for the selected set of database backup volumes? Cross-check the volume history information for this backup series. The reconciliation reports what the self-describing data contains compared to what was learned from the volume history entries. The cross-check involves mounting one or more of the volumes that are indicated by the volume history. Then, using the self-describing data that was included in the database backup volumes, that information is reconciled against what is in the volume history for the database backup. If the information from the volume history file is inconsistent with the self-describing data, then messages are issued to identify the problem. For example, not all values are specified and available, and no self-describing data is found.

If the volume history information is consistent with self-describing data from the database backup, a message is issued indicating that the database backup can be used for restore processing.

If the volume history information is inconsistent with the self-describing data from the database backup or if the self-describing data for the backup cannot be found, error messages are issued indicating what was checked and what was missing.

If the **PREVIEW** parameter is not specified or if it is set to **NO**, and if the volume history and self-describing data from the database backup are consistent, then the restore proceeds.

If the **PREVIEW** parameter is not specified or if it is set to **NO**, and the reconciliation and validation fail, the database restore is not performed. Make extra volumes available and referred to from the volume history file, or remove the incomplete backup series or operation so that the Tivoli Storage Manager server selects a different preferred series or operation and continues processing.

If the **PREVIEW** parameter is set to **YES**, the process performs only the evaluation of the volume history file and the reconciliation and validation against the selected database backup.

### **Example: Restore the database to a specific point in time**

Restore the database to its state on May 12, 2011 at 2:25 PM.

```
dsmserv restore db todate=05/12/2011 totime=14:45
```

### **Example: Restore the server master key without restoring the database**

Restore the server master key without restoring the database by issuing the following command:

```
dsmserv restore db restorekeys=only
```



---

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

You can write 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.

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 472. Return codes

| Return code     | Severity | Numeric value | Description                                                                                                   |
|-----------------|----------|---------------|---------------------------------------------------------------------------------------------------------------|
| RC_OK           | OK       | 0             | The command completed successfully.                                                                           |
| RC_UNKNOWN      | ERROR    | 2             | The command is not found; not a known command.                                                                |
| RC_SYNTAX       | ERROR    | 3             | The 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 command failed because 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_NOTFOUND     | WARNING  | 11            | Returned by a QUERY or SQL SELECT command when no objects are found that match specifications.                |
| 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.                                   |

Table 472. Return codes (continued)

| Return code     | Severity | Numeric value | Description                                                                                     |
|-----------------|----------|---------------|-------------------------------------------------------------------------------------------------|
| RC_IOERROR      | ERROR    | 15            | The command failed because an input/output (I/O) error was encountered on the server.           |
| 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.                                   |
| RC_INVALIDUSE   | ERROR    | 27            | The command failed because the command is used in an invalid manner.                            |
| RC_NOTABLE      | ERROR    | 28            | The command failed because of an unknown SQL table name.                                        |
| RC_FS_NOTCAP    | ERROR    | 29            | The command failed because of incompatible file space name types.                               |
| RC_INVALIDADDR  | ERROR    | 30            | The command failed because of an incorrect high-level address or low-level address.             |
| RC_INVALIDCG    | ERROR    | 31            | The command failed because the management class does not have an archive copy group.            |
| RC_OVERSIZE_VOL | ERROR    | 32            | The command failed because the volume size exceeds the maximum allowed.                         |
| RC_DEFVOL_FAIL  | ERROR    | 33            | The command failed because volumes cannot be defined in RECLAMATIONTYPE=SNAPLOCK storage pools. |
| RC_DELVOL_FAIL  | ERROR    | 34            | The command failed because volumes cannot be deleted in RECLAMATIONTYPE=SNAPLOCK storage pools. |
| RC_CANCELED     | WARNING  | 35            | The command is canceled.                                                                        |

Table 472. Return codes (continued)

| <b>Return code</b> | <b>Severity</b> | <b>Numeric value</b> | <b>Description</b>                                                              |
|--------------------|-----------------|----------------------|---------------------------------------------------------------------------------|
| RC_INVPOLICY       | ERROR           | 36                   | The command failed because there is an invalid definition in the policy domain. |
| RC_INVALIDPW       | ERROR           | 37 <sup>1</sup>      | The command failed because of an invalid password.                              |
| RC_UNSUPP_PARM     | WARNING         | 38 <sup>1</sup>      | The command failed because the command or the parameter is not supported.       |



---

## Appendix B. Accessibility features for the Tivoli Storage Manager product family

Accessibility features help users who have a disability, such as restricted mobility or limited vision to use information technology products successfully.

### Accessibility features

The IBM Tivoli Storage Manager family of products includes the following accessibility features:

- Keyboard-only operation using standard operating-system conventions
- Interfaces that support assistive technology such as screen readers

The command-line interfaces of all products in the product family are accessible.

Tivoli Storage Manager Operations Center provides the following additional accessibility features when you use it with a Mozilla Firefox browser on a Microsoft Windows system:

- Screen magnifiers and content zooming
- High contrast mode

The Operations Center and the Tivoli Storage Manager server can be installed in console mode, which is accessible.

The Operations Center help system is enabled for accessibility. For more information, click the question mark icon on the help system menu bar.

### Vendor software

The Tivoli Storage Manager product family 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.

### IBM and accessibility

See the IBM Human Ability and Accessibility Center (<http://www.ibm.com/able>) for information about the commitment that IBM has to accessibility.

---

1. These return codes are supported in Tivoli Storage Manager Version 6 but are not supported in Version 5.





---

## 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, MD-NC119  
Armonk, NY 10504-1785  
United States of America*

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:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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 want 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  
US*

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 document 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.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

Portions of this code are derived from IBM® Corp. Sample Programs.

© Copyright IBM® Corp. \_enter the year or years\_. All rights reserved.

## **Trademarks**

IBM, the IBM logo, and [ibm.com](http://ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linear Tape-Open, LTO, and Ultrium are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

## **Terms and conditions for product documentation**

Permissions for the use of these publications are granted subject to the following terms and conditions.

### **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### **Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

**Rights** Except as expressly granted in this permission, no other permissions,

licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## **Privacy policy considerations**

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

---

## **Glossary**

A glossary is available with terms and definitions for the IBM Tivoli Storage Manager family of products.

See Tivoli Storage Manager glossary.

To view glossaries for other IBM products, see IBM Terminology.



---

# Index

## Numerics

- 3494SHARED option 1465
- 349X
  - DEFINE LIBRARY command 234
  - UPDATE LIBRARY command 1298
- 3590 device type 162
- 3592 device type 166, 1234
- 3592 drives and media
  - logical block protection 166, 1234
- 4MM device type 172
- 8MM device type 176

## A

- ACCEPT DATE command 30
- ACCESS parameter 383, 1439
- accessibility features 1617
- account record, setting 1080
- ACSACCESSID option 1466
- ACSLCKDRIVE option 1467
- ACSL 1460
- ACSL library 237, 1300
- ACSQUICKINIT option 1468
- ACSTIMEOUTX option 1469
- action, restore 135, 290, 1359
- ACTIVATE POLICYSET command 31
- activating policy sets 31
- active log 1590
- active log mirror 1590
- active-data pool
  - defining a new active-data pool 369
  - identifying with a QUERY command 935
  - restoring data
    - restoring storage pools 1050
    - restoring volumes 1054
  - specifying as backups for primary storage pools 339, 357, 1402, 1418
  - updating an existing active-data pool 1427
- active-data pools
  - list 728
- ACTIVELOGDIRECTORY option 1470
- ACTIVELOGSIZE option 1471
- activity log
  - querying 667
  - setting retention period 1081
  - setting size limit 1081
- ADMINCOMMTIMEOUT option 1472
- ADMINIDLETIMEOUT option 1473
- administrative client
  - batch mode 1, 3
  - console mode 1, 2
  - continuation characters 13
  - interactive mode 1, 3
  - mount mode 1, 3
  - options 5
  - privilege classes 20
  - starting 2
  - stopping 2
  - using 1
- administrative client options
  - alwaysprompt 5, 6
  - commadelimited 6
  - consolemode 6
  - dataonly 6
  - displaymode 6
  - id 6
  - itemcommit 6
  - mountmode 6
  - newlineafterprompt 6
  - noconfirm 6
  - outfile 2, 3, 7
  - password 7
  - quiet 7
  - serveraddress 7
  - tabdelimited 7
  - tcpport 7
  - tcpserveraddress 7
- administrative command
  - background processing 16
  - components 9
  - entry rules 9
  - foreground processing 16
  - help 561
  - issued by all administrators 26
  - issued with operator privilege 25
  - issued with policy privilege 23
  - issued with storage privilege 24
  - naming conventions 14
  - parameters, entering 11
  - privilege class for 20
  - scheduling 300
  - system privilege 20
  - using 29
  - using wildcard characters 14
- administrative commands
  - LOCK ADMIN 601
  - QUERY ADMIN 673
  - REGISTER ADMIN 983
  - REMOVE ADMIN 1008
  - RENAME ADMIN 1017
  - UNLOCK ADMIN 1196
  - UPDATE ADMIN 1207
- administrative schedule
  - associating with profile 273
  - copying 108
  - defining 300
  - deleting 434
  - distributing 279
  - updating 1354
- administrative session, canceling 78
- administrator
  - associating with profile 273
  - distributing 279
  - exporting 483
  - importing 568
- ADMINONCLIENTPORT option 1474
  - administrative sessions using TCPPOINT 1474
  - using with TCPADMINPORT 1568
- affect performance, raw partitions 330

- alert triggers
  - loading 598
- ALIASHALT server option 1475
- ALLOWREORGINDEX option 1476
- ALLOWREORGTABLE option 1477
- ANR9999D error message 1104
- ANS1082E message 137, 292, 1361
- ANS1102E message 137, 292, 1361
- ARCHDELETE parameter 994, 1325
- ARCHFAILOVERLOGDIRECTORY option 1478
- archive
  - defining 154
  - description of 154
  - frequency 155
  - retention grace period 223, 1286
- archive copy group
  - attributes of 154
  - defining 154
  - serialization 154
- archive file deletion by a client 994, 1325
- archive log failover directory 1590
- ARCHLOGCOMPRESS option 1479
- ARCHLOGDIRECTORY option 1480
- ARCHLOGUSEDTHRESHOLD option 1481
- ASSIGN DEFMGMTCLASS command 33
- assigning
  - client node to domain 989, 1320
  - default management class 33
- ASSISTVCRRECOVERY option 1482
- association of client with schedule
  - defining 127
  - deleting 390
  - querying 685
- association, object with profile
  - defining 273
  - deleting 427
- AUDIT LDAPDIRECTORY command 36
- AUDIT LIBRARY command 38
- AUDIT LIBVOLUME command 40
- AUDIT LICENSES command 42
- audit occupancy
  - displaying storage pool utilization 687
  - querying 687
- AUDIT VOLUME command 43
- auditing
  - library 38
  - storage pool volume 43
  - tape volume 40
- AUDITSTORAGE option 1483
- authentication
  - password 1097
  - setting open or closed 1097
- authority
  - client access 554
  - client owner 554
  - for command use 20
  - granting 554
  - QUERYAUTH option 1539
  - REQSYSAUTHOUTFILE option 1547
  - revoking 1058
- auto-update 1169
- AUTOMIGNONUSE parameter 258, 1318

## B

- BACKDELETE parameter 994, 1325

- background process
  - canceling 73
  - query process 73, 847
- backup
  - backup file deletion 994, 1325
  - client 149
  - database 50
  - device information 55
  - primary storage pool 62
  - scheduling for client 286
  - volume history 66
- backup copy group
  - attributes of 149
  - defining 149
  - serialization 149
  - TOCDestination parameter 152
  - updating 1220
- BACKUP DB command 50
- BACKUP DEVCONFIG command 55
- BACKUP NODE command 57
- backup set
  - defining 129
  - deleting 392
  - displaying contents of 695
  - generating 542
  - node groups
    - defining a group 260
    - defining a member 261
    - deleting a group 422
    - deleting a member 423
    - querying a group 832
    - updating a group 1338
  - querying 690
  - updating retention period 1211
- BACKUP STGPOOL command 62
- BACKUP VOLHISTORY command 66
- batch mode
  - restrictions 3
  - starting and ending 3
  - using 3
- BEGIN EVENTLOGGING command 68
- BEGINDATE parameter 668, 753
- BEGINREORGTIME option 1542, 1543
- BEGINTIME parameter 668, 753, 814

## C

- CACHE parameter
  - DEFINE STGPOOL, primary random access 336
  - UPDATE STGPOOL, primary random access 1398
- cancel
  - administrative session 78
  - client node session 78
  - expiration process 71
  - mount requests 76
  - process 73
  - restartable restore session 77
- CANCEL EXPIRATION command 71
- CANCEL PROCESS 17
- CANCEL PROCESS command 73
- CANCEL REPLICATION command 75
- CANCEL REQUEST command 76
- CANCEL RESTORE command 77
- CANCEL SESSION command 78
- canceling commands 17
- CAP parameter
  - MOVE DRMEDIA 86, 617, 634



- CD support 213
- CENTERA device type
  - concurrent read/write access to FILE volumes
    - increasing number of client mount points 989, 1320
- Centera storage device
  - authentication and authorization 182, 1250
  - defining a device class 182
  - updating a device class 1250
- changing date and time on the server 30
- characters available for specifying passwords 14
- CHECKIN LIBVOLUME command 79
- checking in
  - library volume 79
  - with labeling 79
- checking out a library volume 86
- CHECKLABEL parameter
  - AUDIT LIBRARY 39
  - CHECKIN LIBVOLUME 83
  - CHECKOUT LIBVOLUME 89
- CHECKOUT LIBVOLUME command 86
- CHECKTAPEPOS option 1484
- CLASSES parameter
  - GRANT AUTHORITY 554
  - REVOKE AUTHORITY 1058
- CLEAN DRIVE command 92
- cleaning a drive, frequency 227
- client access authority 554
- client node
  - deactivating 114
  - decommissioning 118
  - decommissioning virtual machine 121
- client node session
  - canceling 78
  - scheduling 285
- client option 139
- client owner authority 554
- client-server options, setting 1461
- client, backing up subfiles for 1184
- CLIENTDEDUPTXNLIMIT option 1486
- COLLOCATE parameter
  - DEFINE STGPOOL
    - active-data pool 371
    - copy sequential access 362
    - primary sequential access 350
  - UPDATE STGPOOL
    - active-data pool 1428
    - copy sequential access 1421
    - primary sequential access 1411
- collocation
  - group
    - defining 142
    - deleting 399
    - querying 700
    - updating 1218
  - group member
    - defining 144
    - deleting 400
  - specifying for a storage pool
    - active-data pool 371, 1428
    - copy sequential access 362, 1421
    - primary sequential access 350, 1411
- command
  - DELETE ALERTTRIGGER 389
  - QUERY ALERTSTATUS 680
  - UPDATE ALERTTRIGGER 1202
- command line 8
- command output, formatting 4
- command routing
  - defining server groups for 17
  - defining servers for 17
- command scripts
  - copying 112
  - defining 312
  - deleting 438
  - querying 902
  - renaming 1025
  - running 1064
  - updating 1379
- command-line interface
  - administrative client 1
  - Operations Center 1
  - server console 1
  - using 1
- commandQUERY REPLSERVER
  - command 881
- commands 458
  - ACCEPT DATE 30
  - ACTIVATE POLICYSET 31
  - ASSIGN DEFMGMTCLASS 33
  - AUDIT LDAPDIRECTORY 36
  - AUDIT LIBRARY 38
  - AUDIT LIBVOLUME 40
  - AUDIT LICENSES 42
  - AUDIT VOLUME 43
  - BACKUP DB 50
  - BACKUP DEVCONFIG 55
  - BACKUP NODE 57
  - BACKUP STGPOOL 62
  - BACKUP VOLHISTORY 66
  - BEGIN EVENTLOGGING 68
  - CANCEL EXPIRATION 71
  - CANCEL EXPORT 72
  - CANCEL PROCESS 73
  - CANCEL REPLICATION 75
  - CANCEL REQUEST 76
  - CANCEL RESTORE 77
  - CANCEL SESSION 78
  - CHECKIN LIBVOLUME 79
  - CHECKOUT LIBVOLUME 86
  - CLEAN DRIVE 92
  - COMMIT 93
  - COPY ACTIVATEDATA 95
  - COPY CLOPTSET 99
  - COPY DOMAIN 100
  - COPY MGMTCLASS 102
  - COPY POLICYSET 104
  - COPY PROFILE 106
  - COPY SCHEDULE 108
  - COPY SCRIPT 112
  - COPY SERVERGROUP 113
  - DEACTIVATE DATA 114
  - DECOMMISSION NODE 118
  - DECOMMISSION VM 121
  - DEFINE ALERTTRIGGER 124
  - DEFINE ASSOCIATION 127
  - DEFINE BACKUPSET 129
  - DEFINE CLIENTACTION 133
  - DEFINE CLIENTOPT 139
  - DEFINE CLOPTSET 141
  - DEFINE COLLOGROUP 142
  - DEFINE COLLOCMEMBER 144
  - DEFINE COPYGROUP 148
  - DEFINE DATAMOVER 158
  - DEFINE DEVCLASS 161

commands (continued)

DEFINE DOMAIN 223  
 DEFINE DRIVE 225  
 DEFINE EVENTSERVER 230  
 DEFINE GRPMEMBER 231  
 DEFINE LIBRARY 232, 234, 237, 240, 242, 243, 245, 249,  
 250  
 DEFINE MACHINE 253  
 DEFINE MACHNODEASSOCIATION 255  
 DEFINE MGMTCLASS 257  
 DEFINE NODEGROUP 260  
 DEFINE NODEGROUPMEMBER 261  
 DEFINE PATH 262  
 DEFINE PATH - Destination is a drive 263  
 DEFINE PATH - Destination is a library 268  
 DEFINE POLICYSET 271  
 DEFINE PROFASSOCIATION 273  
 DEFINE PROFILE 279  
 DEFINE RECMEDMACHASSOCIATION 281  
 DEFINE RECOVERYMEDIA 283  
 DEFINE SCHEDULE 285  
 DEFINE SCRATCHPADENTRY 310  
 DEFINE SCRIPT 312  
 DEFINE SERVER 315  
 DEFINE SERVERGROUP 322  
 DEFINE SPACETRIGGER 323  
 DEFINE STATUSTHRESHOLD 326  
 DEFINE STGPOOL 330  
 DEFINE SUBSCRIPTION 377  
 DEFINE VIRTUALFSMAPPING 379  
 DEFINE VOLUME 381  
 DELETE ASSOCIATION 390  
 DELETE BACKUPSET 392  
 DELETE CLIENTOPT 397  
 DELETE CLOPTSET 398  
 DELETE COLLOGGROUP 399  
 DELETE COLLOCMEMBER 400  
 DELETE COPYGROUP 403  
 DELETE DATAMOVER 405  
 DELETE DEVCLASS 406  
 DELETE DOMAIN 407  
 DELETE DRIVE 408  
 DELETE EVENT 409  
 DELETE EVENTSERVER 411  
 DELETE FILESPACE 412  
 DELETE GRPMEMBER 416  
 DELETE KEYRING 417  
 DELETE LIBRARY 418  
 DELETE MACHINE 419  
 DELETE MACHNODEASSOCIATION 420  
 DELETE MGMTCLASS 421  
 DELETE NODEGROUP 422  
 DELETE NODEGROUPMEMBER 423  
 DELETE PATH 424  
 DELETE POLICYSET 426  
 DELETE PROFASSOCIATION 427  
 DELETE PROFILE 430  
 DELETE RECMEDMACHASSOCIATION 432  
 DELETE RECOVERYMEDIA 433  
 DELETE SCHEDULE 434  
 DELETE SCRATCHPADENTRY 437  
 DELETE SCRIPT 438  
 DELETE SERVER 439  
 DELETE SERVERGROUP 440  
 DELETE SPACETRIGGER 441  
 DELETE STATUSTHRESHOLD 442  
 DELETE STGPOOL 444

commands (continued)

DELETE SUBSCRIBER 446  
 DELETE SUBSCRIPTION 447  
 DELETE VIRTUALFSMAPPING 448  
 DELETE VOLHISTORY 449  
 DELETE VOLUME 454  
 DISABLE REPLICATION 462  
 DISABLE SESSIONS 463  
 DISMOUNT VOLUME 466  
 DISPLAY OBJNAME 467  
 DSMADMC 1  
 DSMSERV 1586  
 DSMSERV DISPLAY DBSPACE 1589  
 DSMSERV DISPLAY LOG 1590  
 DSMSERV EXTEND DBSPACE 1592  
 DSMSERV FORMAT 1594  
 DSMSERV REMOVEDB 1602  
 DSMSERV RESTORE DB 1605, 1608  
 DSMSERV RUNFILE 1586  
 ENABLE EVENTS 469  
 ENABLE REPLICATION 472  
 ENABLE SESSIONS 473  
 END EVENTLOGGING 475  
 EXPIRE INVENTORY 477  
 EXPORT ADMIN 483  
 EXPORT NODE 490  
     directly to another server 502  
 EXPORT POLICY 512  
 EXPORT SERVER 519  
     directly to another server 530  
     to sequential media 522  
 EXTEND DB 538  
 GENERATE 541  
 GENERATE BACKUPSET 542  
 GENERATE BACKUPSETTOC 551  
 GRANT AUTHORITY 554  
 GRANT PROXYNODE 558  
 HALT 559  
 HELP 561  
 IDENTIFY DUPLICATES 563  
 IMPORT ADMIN 568  
 IMPORT NODE 572  
 IMPORT POLICY 579  
 IMPORT SERVER 582  
 INSERT MACHINE 588  
 ISSUE MESSAGE 589  
 LABEL LIBVOLUME 591  
 LOAD DEFALERTTRIGGERS 598  
 LOCK NODE 603  
 LOCK PROFILE 605  
 MACRO 607  
 MIGRATE STGPOOL 609  
 MOVE DATA 613  
 MOVE DRMEDIA 617  
 MOVE GRPMEMBER 633  
 MOVE MEDIA 634  
 MOVE NODEDATA 642  
 NOTIFY SUBSCRIBERS 652  
 PERFORM LIBACTION 653  
 PING SERVER 657  
 PREPARE 658  
 QUERY ACTLOG 667  
 QUERY ALERTTRIGGER 678  
 QUERY ASSOCIATION 685  
 QUERY AUDITOCCUPANCY 687  
 QUERY BACKUPSET 690  
 QUERY BACKUPSETCONTENTS 695

commands (continued)

QUERY CLOPTSET 698  
 QUERY COLLOGROUP 700  
 QUERY CONTENT 703  
 QUERY COPYGROUP 710  
 QUERY DATAMOVER 715  
 QUERY DB 718  
 QUERY DBSPACE 721  
 QUERY DEVCLASS 723  
 QUERY DIRSPACE 727  
 QUERY DOMAIN 728  
 QUERY DRIVE 731  
 QUERY DRMEDIA 735  
 QUERY DRMSTATUS 744  
 QUERY ENABLED 747  
 QUERY EVENT 749  
 QUERY EVENTRULES 761  
 QUERY EVENTSERVER 764  
 QUERY EXPORT 765  
 QUERY FILESPACE 772  
 QUERY LIBRARY 780  
 QUERY LIBVOLUME 783  
 QUERY LICENSE 786  
 QUERY LOG 790  
 QUERY MACHINE 792  
 QUERY MEDIA 795  
 QUERY MGMTCLASS 801  
 QUERY MONITORSETTINGS 804  
 QUERY MONITORSTATUS 807  
 QUERY MOUNT 811  
 QUERY NASBACKUP 813  
 QUERY NODE 817  
 QUERY NODEDATA 829  
 QUERY OCCUPANCY 834  
 QUERY OPTION 838  
 QUERY PATH 840  
 QUERY POLICYSET 844  
 QUERY PROCESS 847  
 QUERY PROFILE 851  
 QUERY PROXYNODE 854  
 QUERY PVESTIMATE 855  
 QUERY RECOVERYMEDIA 859  
 QUERY REPLICATION 862  
 QUERY REPLNODE 875  
 QUERY REPLRULE 879  
 QUERY REQUEST 883  
 QUERY RESTORE 884  
 QUERY RPFCONTENT 887  
 QUERY RPFFILE 889  
 QUERY SCHEDULE 892  
 QUERY SCRATCHPADENTRY 900  
 QUERY SCRIPT 902  
 QUERY SERVER 905  
 QUERY SERVERGROUP 910  
 QUERY SESSION 912  
 QUERY SHREDSTATUS 917  
 QUERY SPACETRIGGER 919  
 QUERY SSLKEYRINGPW 921  
 QUERY STATUS 922  
 QUERY STATUSTHRESHOLD 932  
 QUERY STGPOOL 935  
 QUERY SUBSCRIBER 948  
 QUERY SUBSCRIPTION 950  
 QUERY SYSTEM 952  
 QUERY TAPEALERTMSG 954  
 QUERY TOC 955  
 QUERY VIRTUALFSMAPPING 958

commands (continued)

QUERY VOLHISTORY 960  
 QUERY VOLUME 968  
 QUIT 975  
 RECLAIM STGPOOL 976  
 RECONCILE VOLUMES 979  
 REGISTER LICENSE 987  
 REGISTER NODE 989  
 REMOVE NODE 1010  
 REMOVE REPLNODE 1012  
 RENAME FILESPACE 1019  
 RENAME NODE 1023  
 RENAME SCRIPT 1025  
 RENAME SERVERGROUP 1026  
 RENAME STGPOOL 1027  
 REPLICATE NODE 1028  
 RESET PASSEXP 1039  
 RESTORE NODE 1044  
 RESTORE STGPOOL 1049  
 RESTORE VOLUME 1053  
 RESUME EXPORT 1041  
 REVOKE AUTHORITY 1058  
 REVOKE PROXYNODE 1062  
 ROLLBACK 1063  
 RUN 1064  
 SELECT 1067  
 SET ACCOUNTING 1080  
 SET ACTLOGRETENTION 1081  
 SET ALERTACTIVEDURATION 1083  
 SET ALERTCLOSEDDURATION 1084  
 SET ALERTEMAIL 1085  
 SET ALERTEMAILFROMADDR 1086  
 SET ALERTEMAILSMTPHOST 1087  
 SET ALERTEMAILSMTPPORT 1088  
 SET ALERTINACTIVEDURATION 1090  
 SET ALERTMONITOR 1091  
 SET ALERTSUMMARYTOADMINS 1089  
 SET ALERTUPDATEINTERVAL 1092  
 SET ARCHIVERETENTIONPROTECTION 1093  
 SET ARREPLRULEDEFAULT 1095  
 SET AUTHENTICATION 1097  
 SET BKREPLRULEDEFAULT 1098  
 SET CLIENTACTDURATION 1101  
 SET CONFIGMANAGER 1102  
 SET CONFIGREFRESH 1103  
 SET CONTEXTMESSAGING 1104  
 SET CPUINFOREFRESH 1105  
 SET CROSSDEFINE 1106  
 SET DBRECOVERY 1107  
 SET DEDUPVERIFICATIONLEVEL 1109  
 SET DEFAULTAUTHENTICATION 1111  
 SET DISSIMILARPOLICIES 1113  
 SET DRMACTIVEDATASTGPOOL 1115  
 SET DRMCHECKLABEL 1116  
 SET DRMCMDFILENAME 1117  
 SET DRMCOPYSTGPOOL 1118  
 SET DRMCOURIERNAME 1119  
 SET DRMDBBACKUPEXPIREDDAYS 1120  
 SET DRMFILEPROCESS 1122  
 SET DRMINSTRPREFIX 1123  
 SET DRMNOTMOUNTABLENAME 1125  
 SET DRMPLANPREFIX 1126  
 SET DRMPLANVPOSTFIX 1128  
 SET DRMPRIMSTGPOOL 1129  
 SET DRMRPFEXPIREDDAYS 1130  
 SET DRMVaultNAME 1132  
 SET EVENTRETENTION 1133

commands (continued)

SET FAILOVERHLADDRESS 1134  
 SET INVALIDPWLIMIT 1135  
 SET LDAPPASSWORD 1137  
 SET LDAPUSER 1138  
 SET LICENSEAUDITPERIOD 1140  
 SET MAXCMDRETRIES 1141  
 SET MAXSCHEDESESSIONS 1142  
 SET MINPWLENGTH 1144  
 SET MONITOREDSEVERGROUP 1145  
 SET MONITORINGADMIN 1146  
 SET NODEATRISKINTERVAL 1147  
 SET PASSEXP 1149  
 SET PRODUCTOFFERING 1151  
 SET QUERYSCHEDEPERIOD 1153  
 SET RANDOMIZE 1154  
 SET REGISTRATION 1156  
 SET REPLRECOVERDAMAGED 1158  
 SET REPLRETENTION 1161  
 SET REPLSERVER 1163  
 SET RETRYPERIOD 1165  
 SET SCHEDMODES 1166  
 SET SCRATCHPADRETENTION 1168  
 SET SERVERHLADDRESS 1169  
 SET SERVERLLADDRESS 1170  
 SET SERVERNAME 1171  
 SET SERVERPASSWORD 1172  
 SET SPREPLRULEDEFAULT 1173  
 SET SSLKEYRINGPW 1175  
 SET STATUSATRISKINTERVAL 1176  
 SET STATUSMONITOR 1178  
 SET STATUSREFRESHINTERVAL 1180  
 SET STATUSSKIPASFAILURE 1182  
 SET SUMMARYRETENTION 1185  
 SET TOCLOADRETENTION 1187  
 SET VMATRISKINTERVAL 1188  
 SETOPT 1190  
 SHRED DATA 1192  
 SUSPEND EXPORT 1194  
 UNLOCK NODE 1198  
 UNLOCK PROFILE 1200  
 UPDATE ALERTSTATUS 1205  
 UPDATE BACKUPSET 1211  
 UPDATE CLIENTOPT 1216  
 UPDATE CLOPTSET 1217  
 UPDATE COLLOGGROUP 1218  
 UPDATE COPYGROUP 1219  
 UPDATE DATAMOVER 1227  
 UPDATE DEVCLASS 1229  
 UPDATE DOMAIN 1286  
 UPDATE DRIVE 1288  
 UPDATE FILESPACE 1292  
 Update LIBRARY 1309  
 UPDATE LIBRARY 1297, 1298, 1300, 1302, 1303, 1304,  
 1306, 1310  
 UPDATE LIBVOLUME 1313  
 UPDATE MACHINE 1315  
 UPDATE MGMTCLASS 1317  
 UPDATE NODE 1320  
 UPDATE NODEGROUP 1338  
 UPDATE PATH 1339  
 UPDATE PATH - Destination is a drive 1340  
 UPDATE PATH - Destination is a library 1344  
 UPDATE POLICYSET 1347  
 UPDATE PROFILE 1349  
 UPDATE RECOVERYMEDIA 1350  
 UPDATE REPLRULE 1352

commands (continued)

UPDATE SCHEDULE 1354  
 UPDATE SCRATCHPADENTRY 1378  
 UPDATE SCRIPT 1379  
 UPDATE SERVER 1382  
 UPDATE SERVERGROUP 1386  
 UPDATE SPACETRIGGER 1387  
 UPDATE STATUSTHRESHOLD 1389  
 UPDATE STGPOOL 1393  
 UPDATE VIRTUALFSMAPPING 1434  
 UPDATE VOLHISTORY 1436  
 UPDATE VOLUME 1438  
 VALIDATE LANFREE 1443  
 VALIDATE POLICYSET 1445  
 VALIDATE REPLICATION 1447  
 VALIDATE REPLPOLICY 1452  
 VARY 1455  
 commands in a macro  
   committing 93  
   rolling back 1063  
 commands, administrative 8  
 commands, canceling 17  
 commandsREMOVE REPLSERVER  
   command 1014  
 COMMIT command 93  
 committing commands in a macro 93  
 COMMMETHOD option 1488  
 COMMTIMEOUT option 1490  
 communications, server-to-server 315  
   COMMMETHOD option 1488  
   shared memory between server and client 1488  
 complex password  
   AUDIT LDAPDIRECTORY command 36  
   SET DEFAULTAUTHENTICATION 1111  
   SET LDAPUSR command 1137  
 configuration manager 377  
 configuration profile 279  
 console mode  
   ending 2  
   restrictions 2  
   using 2  
 continuation characters  
   for a list of values 13  
   for a quoted list of values 13  
   in output file 5  
   using the maximum number of 13  
 conventions  
   typographic xi  
 COPIED parameter, QUERY CONTENT 705  
 COPY ACTIVATEDATA command 95  
 COPY CLOPTSET command 99  
 COPY DOMAIN command 100  
 copy group  
   defining archive 154  
   defining backup 149  
   deleting 403  
   description of 148  
   querying 710  
   restriction 148  
   updating archive 1224  
   updating backup 1220  
 COPY MGMTCLASS command 102  
 COPY POLICYSET command 104  
 COPY PROFILE command 106  
 COPY SCHEDULE command 108  
 COPY SCRIPT 112  
 COPY SERVERGROUP command 113

- COPYCONTINUE parameter
  - DEFINE STGPOOL, primary random access 339
  - DEFINE STGPOOL, primary sequential access 357
  - UPDATE STGPOOL, primary random access 1401
  - UPDATE STGPOOL, primary sequential access 1417
- copying
  - management class 102
  - policy domain 100
  - policy set 104
  - profile 106
  - schedule 108
  - script 112
  - server group 113
- COPYSTGPOOLS parameter
  - DEFINE STGPOOL, primary random access 338
  - DEFINE STGPOOL, primary sequential access 356
  - UPDATE STGPOOL, primary random access 1401
  - UPDATE STGPOOL, primary sequential access 1416
- COUNT parameter, QUERY CONTENT 704
- CRCDATA parameter
  - DEFINE STGPOOL, primary random access 335
  - DEFINE STGPOOL, primary sequential access 346, 367
  - UPDATE STGPOOL, primary random access 1397
  - UPDATE STGPOOL, primary sequential access 1408, 1425
- creating
  - backup set 542
  - client files onto a set of media 542

## D

- damaged
  - files
    - recovering 1028
- damaged files
  - recovering 1158
- DAMAGED parameter, QUERY CONTENT 705
- data
  - moving 613
  - removing expired 477
- data deduplication 358, 367, 368, 376, 1419, 1425, 1432, 1433
- data deduplication verification level, setting 1109
- data mover
  - defining 158
  - deleting 405
  - querying 715
  - updating 1227
- data protection
  - logical block protection 166, 190, 203, 1234, 1269
- data protection using WORM FILE volumes and SnapLock
  - when defining active-data pools 373
  - when defining copy storage pools 364
  - when defining sequential access storage pools 349
- data shredding, storage pools
  - backing up 64
  - defining 340
  - moving data 615
  - updating 1403
- database
  - backup 50
  - extending 538
  - installing 1594
  - querying 718
  - removing 1602
  - restoring 1605, 1608
  - setting options for 1461
  - transfer by data mover 158
- database recoverable space 718
- database recovery
  - back up volume history 66
  - delete volume history 449
  - query volume history 960
- database storage space 721, 1589, 1592
- DATAFORMAT parameter, define primary sequential access
  - storage pool 354, 366
- dataonly 6
- DATES parameter
  - IMPORT NODE 576
  - IMPORT SERVER 584
- DAYOFWEEK parameter
  - UPDATE SCHEDULE
    - client 1366
- DBDIAGLOGSIZE option 1491
- DBDIAGPATHFSTHRESHOLD option 1493
- DBMEMPERCENT option 1494
- DBMTCPPORT option 1495
- DEACTIVATE DATA command 114
- Deactivate data for a Data Protection client node 114
- debug ANR9999D message 1104
- DECOMMISSION NODE command 118
- DECOMMISSION VM command 121
- decommissioning client node 118
- decommissioning virtual machine 121
- deduplicating data 563
- deduplication 358, 367, 368, 376, 1419, 1425, 1432, 1433
- DEDUPTIER2FILESIZE option 1497
- DEDUPTIER3FILESIZE option 1498
- default management class, assigning 33
- Define a 349X library 234
- Define a file library 242
- Define a manual library 243
- Define a SCSI library 245
- Define a shared library 249
- Define a VTL library 250
- DEFINE ALERTTRIGGER command 124
- Define an ACSLS library 237
- DEFINE ASSOCIATION command 127
- DEFINE BACKUPSET command 129
- DEFINE CLIENTACTION command 133
- DEFINE CLIENTOPT command 139
- DEFINE CLOPTSET command 141
- DEFINE COLLOGGROUP command 142
- DEFINE COLLOCMEMBER command 144
- DEFINE COPYGROUP command 148
- DEFINE DATAMOVER command 158
- DEFINE DEVCLASS command 161
- DEFINE DOMAIN command 223
- DEFINE DRIVE command 225
- DEFINE EVENTSERVER command 230
- DEFINE GRPMEMBER command 231
- DEFINE LIBRARY command 232, 234, 237, 240, 242, 243, 245, 249, 250
- DEFINE MACHINE command 253
- DEFINE MACHNODEASSOCIATION command 255
- DEFINE MGMTCLASS command 257
- DEFINE PATH command 262, 263, 268
- DEFINE POLICYSET command 271
- DEFINE PROFASSOCIATION command 273
- DEFINE PROFILE command 279
- DEFINE RECMEDMACHASSOCIATION command 281
- DEFINE RECOVERYMEDIA command 283
- DEFINE SCHEDULE command 285
- DEFINE SCRATCHPADENTRY 310
- DEFINE SCRIPT command 312
- DEFINE SERVER command 315

DEFINE SERVERGROUP command 322  
 DEFINE SPACETRIGGER command 323  
 DEFINE STATUSTHRESHOLD command 326  
 DEFINE STGPOOL command 330  
 DEFINE SUBSCRIPTION command 377  
 DEFINE VIRTUALFSMAPPING command 379  
 DEFINE VOLUME command 381  
 defining
 

- association 127
- backup set 129
- client with schedule 286
- collocation group member 144
- collocation groups 142
- configuration manager 377
- copy group 148
- device class 161
- domain 223
- drive 225
- event server 230
- file library 242
- group member 231
- library 232, 234, 237, 240, 243, 245, 249
- management class 257
- node group 260
- node group member 261
- object with profile 273
- path for NAS data mover 262
- path for NDMP (NAS) connection 262
- policy set 271
- profile 279
- profile association 273
- schedule 285
- script 312
- server 315
- server group 322
- space trigger 323
- storage pool 330
- storage pool volume 381
- subscription 377
- VTL library 250

 DELETE ALERTTRIGGER command 389  
 DELETE ASSOCIATION command 390  
 DELETE BACKUPSET 392  
 DELETE CLIENTOPT command 397  
 DELETE CLOPTSET command 398  
 DELETE COLLOGROUP command 399  
 DELETE COLLOCMEMBER command 400  
 DELETE COPYGROUP command 403  
 DELETE DATAMOVER 405  
 DELETE DEVCLASS command 406  
 DELETE DOMAIN command 407  
 DELETE DRIVE command 408  
 DELETE EVENT command 409  
 DELETE EVENTSERVER command 411  
 DELETE FILESPACE command 412  
 DELETE KEYRING command 417  
 DELETE LIBRARY command 418  
 DELETE MACHINE command 419  
 DELETE MACHNODEASSOCIATION command 420  
 DELETE MGMTCLASS command 421  
 DELETE PATH 424  
 DELETE POLICYSET command 426  
 DELETE PROFASSOCIATION command 427  
 DELETE PROFILE command 430  
 DELETE RECMEDMACHASSOCIATION command 432, 1117  
 DELETE RECOVERYMEDIA command 433  
 DELETE SCHEDULE command 434  
 DELETE SCRATCHPADENTRY 437  
 DELETE SCRIPT command 438  
 DELETE SERVER command 439  
 DELETE SERVERGROUP command 440  
 DELETE SPACETRIGGER command 441  
 DELETE STATUSTHRESHOLD command 442  
 DELETE STGPOOL command 444  
 DELETE SUBSCRIBER command 446  
 DELETE SUBSCRIPTION command 447  
 DELETE VIRTUALFSMAPPING command 448  
 DELETE VOLHISTORY command 449  
 DELETE VOLUME command 454  
 deleting
 

- archive file deletion by a client, allowing 994, 1325
- backup file deletion by a client, allowing 994
- backup file deletion by a client, ing 1325
- backup set 392
- collocation group 399
- collocation group member 400
- copy group 403
- device class 406
- domain 407
- drive 408
- event record 409
- event server 411
- expired data 477
- file space 412
- group member 416
- library 418
- management class 421
- node group 422
- node group member 423
- policy set 426
- profile 430
- profile association 427
- schedule 434
- script 438
- server 439
- server group 440
- space trigger 441
- storage pool 444
- storage pool volume 454
- subscriber 446
- subscription 447
- volume history 449

 deployment
 

- automatic 1169
- determining retention periods for policy domains 728

 DEVCONFIG option 1499  
 device class
 

- 3590 1230
- 3592 166
- CENTERA 182, 1250
- defining 161
- deleting 406
- NAS 210, 1275
- querying 723
- updating 1229
- VOLSAFE 83, 218, 1282

 device configuration file 55  
 device names
 

- for devices connected to NAS file server 262

 device type
 

- 3590 162, 1230
- 3592 166, 1234
- 4MM 172, 1240
- 8MM 176, 1244

- device type *(continued)*
  - CENTERA 182, 1250
  - DLT 184, 1252
  - ECARTRIDGE 190, 1257
  - FILE 197, 727, 1263
  - GENERICTAPE 200, 1267
  - LTO 203, 204, 1269
  - NAS 210, 1275
  - REMOVABLEFILE 213, 1278
  - SERVER 216, 1280
  - VOLSAFE 218, 1282
- directories, specifying multiple for FILE volumes 198, 1264
- disability 1617
- DISABLE EVENTS 458
- DISABLE EVENTS command 458
- DISABLE REPLICATION command 462
- DISABLE SESSIONS command 463
- DISABLEREORGTABLE option 1500
- DISABLESCHEDS option 1501
- disabling
  - context messaging for ANR9999D messages 1104
  - server sessions, inbound and outbound 463
- DISALLOWDES option 1502
- disaster recovery 1158
- disaster recovery media
  - moving offsite and onsite 617
- DISCARDDATA parameter 455
- disk space, migrating data to create 335, 609, 1398
- disk volume performance 330
- disk volumes 330
- disk-only backup
  - defining FILE device classes 198
  - defining volumes 381
  - updating FILE device classes 1264
- DISMOUNT VOLUME command 466
- dismounting volumes 466
- DISPLAY OBJNAME command 467
- displaying
  - full object name 467
  - output 6
  - product version information 6
  - scheduled and completed events 749
- DISPLAYLFINFO option 1503
- DLT
  - device type 184
  - WORM media 184
- DNSLOOKUP option 1504
- domain
  - associating with profile 273
  - copy 100
  - defining 223
  - deleting 407
  - distributing 279
  - querying 728
  - updating 1286
- drive
  - cleaning 92, 227
  - defining 225, 731
  - deleting 408
  - querying 731
  - updating 1288
- drive encryption
  - 3592 device class 166, 1234
  - ECARTRIDGE device class 190, 1257
  - LTO device class 203, 1269
- DRIVEACQUIRERETRY option 1505

- DRIVEENCRYPTION parameter
  - 3592 device class 166, 1234
  - ECARTRIDGE device class 190, 1257
  - LTO device class 203, 1269
- DSMADMC command 1
- DSMADMC options 5
- DSMSERV command 1586
- DSMSERV DISPLAY DBSPACE command 1589
- DSMSERV DISPLAY LOG command 1590
- DSMSERV EXTEND DBSPACE command 1592
- DSMSERV FORMAT command 1594
- DSMSERV REMOVEDB command 1602
- DSMSERV RESTORE DB command 1605, 1608
- DSMSERV RUNFILE command 1586
- duplicate-identification processes 358, 368, 376, 1419, 1425, 1433
- DURATION parameter
  - DEFINE SCHEDULE, administrative command 305
  - DEFINE SCHEDULE, client 295
  - SHRED DATA, administrative command 1192
  - UPDATE SCHEDULE, administrative command 1373
  - UPDATE SCHEDULE, client 1364

## E

- ECARTRIDGE device type 190
- ENABLE EVENTS command 469
- ENABLE REPLICATION command 472
- ENABLE SESSIONS command 473
- ENABLENASDEDUP option 1506
- enabling
  - context messaging for ANR9999D 1104
  - server sessions, inbound and outbound 473
  - target replication server policies 1113
- encryption
  - application
    - 3592 166, 1234
    - ECARTRIDGE 190, 1257
    - LTO 203, 1269
  - drive
    - 3592 166, 1234
    - ECARTRIDGE 190, 1257
    - LTO 203, 1269
  - library 166, 1234
  - system 166, 1234
- END EVENTLOGGING command 475
- ENDDATE parameter 669, 753, 814
- ending
  - batch mode 3
  - interactive mode 2
- ENDTIME parameter 669, 753, 815
- error
  - ANR9999D message 1104
  - file specification 137, 292, 1361
  - In Progress status for QUERY EVENT 755
- ESTCAPACITY parameter
  - DEFINE DEVCLASS
    - 3590 162
    - 3592 166
    - 4MM 172
    - 8MM 176
    - DLT 184
    - ECARTRIDGE 190
    - GENERICTAPE 200
    - LTO 203
    - NAS 210
    - VOLSAFE 218

ESTCAPACITY parameter (*continued*)

UPDATE DEVCLASS

3590 1230

3592 1234

4MM 1240

8MM 1244

DLT 1252

ECARTRIDGE 1257

GENERICTAPE 1267

NAS 1275

VOLSAFE 1282

event logging

send events to receivers 68

set logging off by receivers 475

event record

description of 409, 1133

managing 1133

querying 749

removing from the database 409

setting retention period 1133

event record retention period

managing 1133

querying 922

event, querying 749

EVENTSERVER option 1507

EXCEPTIONSONLY parameter 753

EXPINTERVAL option 1508

EXPIRATION parameter

DEFINE SCHEDULE, administrative command 300

DEFINE SCHEDULE, client 286

UPDATE SCHEDULE, administrative command 1368

UPDATE SCHEDULE, client 1355

EXPIRE INVENTORY command 477

expiring inventory 477

export

administrator 483

node 490

policy information 512

server 519

EXPORT ADMIN

information directly to another server 488

to sequential media 485

EXPORT ADMIN command 483

EXPORT NODE

export node definitions to sequential media 493

EXPORT NODE command 490

EXPORT POLICY command 512

EXPORT SERVER command 519

EXPQUIET option 1509

EXTEND DB command 538

extending

database 538

## F

failover directory for the archive log 1590

FFDCLOGNAME option 1510

FFDCMAXLOGSIZE option 1511

FFDCNUMLOGS option 1512

FILE device type

concurrent read/write access to FILE volumes

increasing number of client mount points 989, 1320

visible as output in QUERY MOUNT 811

visible as output in QUERY SESSION 912

creating and formatting private volumes 381

defining the device class 197

updating the device class 1263

file server, NAS

names for connected devices 262

node 995

file space

automatic rename for Unicode support 1326

deleting 412

FSID 1020

hexadecimal representation 1020

querying 772

renaming 1019

specification errors, restrictions 135, 290, 1359

FILEDATA parameter

EXPORT NODE 495, 504

IMPORT NODE 574

FILEEXIT option 1513

FILENAMES parameter

BACKUP DEVCONFIG 55

BACKUP VOLHISTORY 67

files

collocating by client 350, 362, 371

damaged

recovering 989, 1028, 1158

moving 613

removing expired 477

FILESPEC parameter

EXPORT NODE 494, 503

IMPORT NODE 574

FILETEXTEXIT option 1514

firewall, opening server ports

administrative sessions 1568

client sessions 1571

TCP/IP ports 1568, 1571

FORCE parameter, CHECKOUT LIBVOLUME 90

FORMAT parameter

DEFINE DEVCLASS

3590 162

3592 166

4MM 172

8MM 176

DLT 184

ECARTRIDGE 190

LTO 203

VOLSAFE 218

UPDATE DEVCLASS

3590 1230

3592 1234

4MM 1240

8MM 1244

DLT 1252

ECARTRIDGE 1257

LTO 1269

VOLSAFE 1282

FORMAT, DSMSEV 1594

FREQUENCY parameter

DEFINE COPYGROUP, archive 155

DEFINE COPYGROUP, backup 150

UPDATE COPYGROUP, archive 1224

UPDATE COPYGROUP, backup 1221

FSUSEDTHRESHOLD option 1515

full object name

displaying 467

## G

GENERATE BACKUPSET command 542

GENERATE BACKUPSETTOC 551

GENERICTAPE device type 200, 1267



- globally unique identifier (GUID) 817
- GRANT AUTHORITY command 554
- GRANT PROXYNODE command 558
- granting
  - authority 554
  - types of privilege classes 554
- group, collocation by
  - defining a group 142
  - defining a group member 144
  - deleting a group 399
  - deleting a group member 400
  - querying a group 700
  - updating a group 1218
- group, node
  - defining 260
  - defining a member 261
  - deleting a group 422
  - deleting a member 423
  - querying a group 832
  - updating a group 1338
- GUID 817

## H

- HALT command 559
- halting the server 559
- HELP command 561
- help for administrative commands 561
- hierarchical storage management
  - DEFINE MGMTCLASS 257
  - description of 257
  - UPDATE MGMTCLASS 1317
- hierarchy of storage pools
  - considerations for backup 62
  - defining 330
- high level address, Centera 182, 1250
- HIGHMIG parameter
  - DEFINE STGPOOL
    - primary random access 335
    - primary sequential access 347
  - UPDATE STGPOOL
    - primary random access 1398
    - primary sequential access 1409
- history information
  - backing up 66
  - deleting 449
  - querying 960
- HSM (hierarchical storage management) 258, 1317

## I

- IBM Knowledge Center xi
- IDENTIFY DUPLICATES command 563
- IDLETIMEOUT option 1516
- IMPORT ADMIN command 568
- IMPORT NODE command 572
- IMPORT POLICY command 579
- IMPORT SERVER command 582
- importing
  - administrator 568
  - node 572
  - policy information 579
  - server 582
- In Progress status for QUERY EVENT command 755
- incremental backup of database 50

- inheritance, copy storage pool lists and
  - defining random access storage pools 338
  - defining sequential access storage pools 356
  - updating random access storage pools 1401
  - updating sequential access storage pools 1417
- INSERT MACHINE command 588
- installing
  - database 1594
  - recovery log 1594
- interactive mode
  - continuation characters 13
  - ending 2
  - quitting 975
  - restrictions 3
  - using 3
- inventory expiring 477
- IP address, Centera 182, 1250
- ISSUE MESSAGE command 589
- issuing commands for a one-time action 133
- ITEMCOMMIT option 6

## K

- KEEPALIVE option 1517
- keepalive, TCP
  - enabling 1532
- KEEPALIVEINTERVAL option 1519
- KEEPALIVETIME option 1518
- keyboard 1617
- Knowledge Center xi

## L

- label
  - for REMOVABLEFILE device type 225
  - library volume 591
- LAN
  - DEFINE LIBRARY command 240, 1302
- LAN-free data movement
  - validating 1443
- language limitations 1
- LANGUAGE option 1520
- LBPROTECT parameter 166, 190, 203, 1234, 1269
- LDAP-authenticated password
  - LOCK NODE command 603
  - SET LDAPUSER command 1138
- LDAPCACHEDURATION server option 1522
- LDAPURL server option 1523
- library
  - auditing 38
  - defining 232, 234, 237, 240, 243, 245, 249
  - defining file 242
  - deleting 418
  - querying 780
  - shared 249, 1309
  - updating 1297, 1298, 1300, 1302, 1304, 1306, 1309
  - updating file 1303
- library volume
  - checking in 79
  - checking out 86
  - labeling 591
  - querying 783
  - updating 1313
- license
  - auditing 42
  - querying 786

- license (*continued*)
  - registering 987
  - setting audit period 1140
- LOAD DEFALERTTRIGGERS command 598
- loading
  - alert triggers 598
- local area network (LAN)
  - DEFINE LIBRARY command 240, 1302
- LOCK ADMIN command 601
- LOCK NODE command 603
- LOCK PROFILE command 605
- locking
  - administrator 601
  - node 603
  - profile 605
- logic flow statements in scripts 1613
- logical block protection 166, 190, 203, 1234, 1269
- logs, recovery 1590
- LOWMIG parameter
  - DEFINE STGPOOL
    - primary random access 335
    - primary sequential access 348
  - UPDATE STGPOOL
    - primary random access 1398
    - primary sequential access 1410
- LTO
  - device class
    - defining 203, 204
    - updating 1269
  - WORM 203
- LTO Ultrium drives and media
  - logical block protection 203, 1269

## M

- MACRO command 607
- macros
  - continuation characters 13
  - rolling back 1063
  - using 607
- maintenance 1169
- maintenance mode 1586
- managed server 377
- managed system for storage area network (SAN)
  - DEFINE LIBRARY command 232
  - UPDATE LIBRARY command 1297
- management class
  - copying 102
  - defining the default 33
  - deleting 421
  - querying 801
  - updating 1317
- MAXCAPACITY parameter
  - DEFINE DEVCLASS
    - FILE 198
    - REMOVABLEFILE 213
    - SERVER 216
  - UPDATE DEVCLASS
    - FILE 1263
    - REMOVABLEFILE 1278
    - SERVER 1280
- maximum retries, setting 1141
- MAXPROCESS parameter
  - BACKUP STGPOOL 63
  - UPDATE STGPOOL 1051

- MAXSCRATCH parameter
  - DEFINE STGPOOL
    - active-data pool 374
    - copy sequential access 365
    - primary sequential access 351
  - UPDATE STGPOOL
    - active-data pool 1431
    - copy sequential access 1424
    - primary sequential access 1413
- MAXSESSIONS option 1524
- MAXSIZE parameter
  - DEFINE STGPOOL
    - primary random access 334
    - primary sequential access 346
  - UPDATE STGPOOL
    - primary random access 1396
    - primary sequential access 1407
- media
  - moving offsite and onsite 617
- media mount requests 76
- media support, CD 213
- member, server group 231
- MERGEFILESPPACES parameter
  - EXPORT NODE 509
  - EXPORT SERVER 536
  - IMPORT NODE 577
  - IMPORT SERVER 582
- message diagnosis, ANR9999D 1104
- MESSAGEFORMAT option 1525
- MIGDESTINATION parameter
  - DEFINE MGMTCLASS 258
  - UPDATE MGMTCLASS 1318
- MIGPROCESS parameter
  - DEFINE STGPOOL, primary random access 336
  - UPDATE STGPOOL, primary random access 1398
- MIGRATE STGPOOL command 609
- migration
  - files from client node 258
  - specifying multiple concurrent processes
    - using the DEFINE STGPOOL command 353
    - using the UPDATE STGPOOL command 1415
  - starting manually 609
  - storage pool, random access
    - high migration threshold 335, 1398
    - low migration threshold 335, 609, 1398
  - storage pool, sequential access
    - high migration threshold 347, 1409
    - low migration threshold 348, 609, 1410
- MIGREQUIRESBKUP parameter
  - DEFINE MGMTCLASS 258
  - UPDATE MGMTCLASS 1318
- MIRRORLOGDIRECTORY option 1526
- MODE parameter
  - DEFINE COPYGROUP, archive 156
  - DEFINE COPYGROUP, backup 151
  - UPDATE COPYGROUP, archive 1225
  - UPDATE COPYGROUP, backup 1222
- monitoring
  - Tivoli Storage Manager activities 3
- monitoring Tivoli Storage Manager activities 2
- mount mode
  - ending 3
  - using 3
- mount request 76
  - canceling 76
  - querying 883

- mounted sequential access volumes
  - dismounting 466
  - querying 811
- MOUNTLIMIT parameter
  - DEFINE DEVCLASS
    - 3590 162
    - 3592 166
    - 4MM 172
    - 8MM 176
    - CENTERA 183
    - DLT 184
    - ECARTRIDGE 190
    - GENERICTAPE 200
    - LTO 203
    - NAS 210
    - REMOVABLEFILE 213
    - SERVER 217
    - VOLSAFE 218
  - UPDATE DEVCLASS
    - 3590 1230
    - 3592 1234
    - 4MM 1240
    - 8MM 1244
    - CENTERA 1251
    - DLT 1252
    - ECARTRIDGE 1257
    - FILE 1263
    - GENERICTAPE 1267
    - LTO 1269
    - REMOVABLEFILE 1278
    - VOLSAFE 1282
- WORM devices and media
  - 8mm devices 176
  - Sony AIT50/AIT100, supported in 8mm class definition 176
- MOUNTRETENTION parameter
  - DEFINE DEVCLASS
    - 3590 162
    - 3592 166
    - 4MM 172
    - 8MM 176
    - DLT 184
    - ECARTRIDGE 190
    - GENERICTAPE 200
    - LTO 203
    - NAS 210
    - REMOVABLEFILE 213
    - SERVER 216
    - VOLSAFE 218
  - UPDATE DEVCLASS
    - 3590 1230
    - 3592 1234
    - 4MM 1240
    - 8MM 1244
    - DLT 1252
    - ECARTRIDGE 1257
    - GENERICTAPE 1267
    - LTO 1269
    - NAS 1275
    - REMOVABLEFILE 1278
    - SERVER 1280
    - VOLSAFE 1282
- MOUNTWAIT parameter
  - DEFINE DEVCLASS
    - 3590 162
    - 3592 166
    - 4MM 172

- MOUNTWAIT parameter *(continued)*
  - DEFINE DEVCLASS *(continued)*
    - 8MM 176
    - DLT 184
    - ECARTRIDGE 190
    - GENERICTAPE 200
    - LTO 203
    - NAS 210
    - REMOVABLEFILE 213
    - VOLSAFE 218
  - UPDATE DEVCLASS
    - 3590 1230
    - 3592 1234
    - 4MM 1240
    - 8MM 1244
    - DLT 1252
    - ECARTRIDGE 1257
    - GENERICTAPE 1267
    - LTO 1269
    - NAS 1275
    - REMOVABLEFILE 1278
    - SERVER 1281
    - VOLSAFE 1282
- MOVE DATA command 613
- MOVE DRMEDIA command 617
- MOVE GRPMEMBER command 633
- MOVE MEDIA command 634
- MOVE NODEDATA command 642
- MOVEBATCHSIZE option 1527
- MOVESIZETHRESH option 1528
- moving
  - data in storage pool 613
  - files 613
  - group member 633
- MSGINTERVAL option 1529
- MSGNO parameter, QUERY ACTLOG 670

## N

- name
  - devices 262
  - for device connected to NAS file server 262
- naming conventions
  - introduction 14
  - naming volumes 14
  - passwords 13
  - restrictions 14
- NAS file server
  - data mover, defining 158
  - name for connected device 262
  - node name 158
  - path, defining 262
- NDMP operations for NAS file servers
  - prevent closing of inactive connections
    - enabling TCP keepalive 1532
    - specifying connection timeout 1530
- NDMPCONNECTIONTIMEOUT server option 1530
- NDMPCONTROLPORT option 1531
- NDMPENABLEKEEPALIVE server option 1532
- NDMPPORTRANGE option 1533
- NDMPREFDATAINTERFACE option 1534
- NetApp DataONTAP, for managing FILE volumes
  - when defining active-data pools 373
  - when defining copy storage pools 364
  - when defining sequential access storage pools 349
- network-attached storage (NAS) file server
  - data mover, defining 158

network-attached storage (NAS) file server *(continued)*

device class 210, 1275

path, defining 262

NEXTSTGPOOL parameter

DEFINE STGPOOL

primary random access 335

primary sequential access 347

UPDATE STGPOOL

primary random access 1397

primary sequential access 347, 1409

node

associating with a schedule 127

exporting 490

finding file for 834

importing 572

locking 603

name for data mover 158

NAS file server 995

node group

defining a group 260

defining a member 261

deleting a group 422

deleting a member 423

querying a group 832

updating a group 1338

querying 817

registering 989

removing 1010

renaming 1023

unlocking 1198

updating 1320

node replication

configuration, measuring the effectiveness of 875

configuration, validating 1447

disabling 462

disabling and enabling

all client nodes, replication for 463, 473

data types in file spaces 1292

individual nodes, replication for 989, 1320

replication rules 1352

enabling 472

failover address

setting 1134

file spaces

data types, disabling and enabling 1292

purging data in 1292

querying replication results 772, 862

rules, changing 1292

nodes

adding for replication 989, 1320

individual, disabling and enabling replication for 989, 1320

removing from replication 1012

system-wide, disabling and enabling replication for 463

systemwide, disabling and enabling replication for 473

previewing results 1028

process information

displaying 772, 862

record retention 922, 1161

records, setting retention for 1161

removing client node 1012

replicating

canceling processes 75

data by command 1028

data by type, priority, and file space 1028

throughput, managing 1028

node replication *(continued)*

results, previewing 1028

rules

client node 817, 989, 1320

disabling and enabling 1352

file space 772, 1292

querying 772, 817, 879

server 1095, 1098, 1173

Secure Sockets Layer (SSL) 315, 905, 1382

synchronizing exported-imported data 989, 1320

target replication server

changing 1163

setting 1163

validating a configuration 1447

verifying

differences between policies 1452

NOPREEMPT option 1535

NORETRIEVEDATE option 1536

normal recovery log mode 50

NOTIFY SUBSCRIBERS command

example 652

related commands 652

NUMOPENVOLSALLOWED option 1537

## O

OBJECTS parameter

DEFINE SCHEDULE, client 295

UPDATE SCHEDULE, client 1364

occupancy

displaying file space information 834

querying 834

offsite volumes, specifying the number to be reclaimed

using the DEFINE STGPOOL command 365, 373

using the UPDATE STGPOOL command 1423, 1430

Operations Center 8

option set 139

OPTIONS parameter

DEFINE SCHEDULE, client 295

UPDATE SCHEDULE, client 1364

options, client 139

options, server

importance for recovery 50

querying 838

output

displaying 6

output headers

displaying 6

overflow location for logs 1590

OWNER parameter, DELETE FILESPACE 414

## P

parameters, descriptions in 15

parameters, entering 11

password

case-sensitivity 13, 14

characters allowed for entering 13

entered with the REGISTER ADMIN command 983

entered with the REGISTER NODE command 989

entered with the UPDATE ADMIN command 1207

setting expiration interval 1149

setting the maximum length 14

path

defining 262

defining to a drive 263

- path *(continued)*
  - defining to a library 268
  - deleting 424
  - querying 840
  - updating 1339
  - updating to a drive 1340
  - updating to a library 1344
- pattern-matching expression, creating 14
- PERFORM LIBACTION command 653
- performance
  - improving communications with client using shared memory 1488, 1554
  - limiting offsite volumes to be reclaimed
    - using the DEFINE STGPOOL command 365, 373
    - using the UPDATE STGPOOL command 1423, 1430
  - specifying multiple concurrent migration processes
    - using the DEFINE STGPOOL command 353
    - using the UPDATE STGPOOL command 1415
  - specifying multiple concurrent reclamation processes
    - using the DEFINE STGPOOL command 364, 372
    - using the UPDATE STGPOOL command 1423, 1430
  - storage requirements, reducing by deduplication 358, 367, 368, 376, 1419, 1425, 1432, 1433
- PERUNITS parameter
  - DEFINE SCHEDULE, administrative command 308
  - UPDATE SCHEDULE, administrative command 1376
  - UPDATE SCHEDULE, client 1366
- policies
  - enabling 1113
- policy domain
  - associating with profile 273
  - copy 100
  - defining 223
  - deleting 407
  - distributing 279
  - querying 728
  - updating 1286
- policy privilege class
  - privilege class
    - policy 23
  - restricted 23
  - unrestricted 23
- policy set
  - activating 31
  - copying 104
  - defining 271
  - deleting 426
  - querying 844
  - updating 1347
  - validating 1445
- polling
  - information about 1153
  - setting frequency 1153
- Pool Entry Authorization (PEA) file, Centera 182, 1250
- ports for firewall
  - administrative sessions 1568
  - client sessions 1571
  - TCP/IP ports 1568, 1571
- preempting of operations 1535
- PREFIX parameter
  - DEFINE DEVCLASS
    - 3590 162
    - 3592 166
    - 4MM 172
    - 8MM 176
    - DLT 184
    - ECARTRIDGE 190
  - PREFIX parameter *(continued)*
    - DEFINE DEVCLASS *(continued)*
      - LTO 203
      - NAS 210
      - SERVER 216
      - VOLSAFE 218
    - UPDATE DEVCLASS
      - 3590 1230
      - 3592 1234
      - 4MM 1240
      - 8MM 1244
      - DLT 1252
      - ECARTRIDGE 1257
      - LTO 1269
      - NAS 1275
      - SERVER 1280
      - VOLSAFE 1282
    - PREPARE command 658
    - PREVIEW parameter
      - EXPORT NODE 495, 575
    - PREVIEWIMPORT parameter
      - EXPORT ADMIN 488
      - EXPORT NODE 509
      - EXPORT POLICY 517
      - EXPORT SERVER 536
    - printing redirected output 4
    - privilege class
      - administrator issued 26
      - none required 26
      - operator 25
      - policy 24
      - restricted 24
      - storage 24, 25
      - system 20
      - unrestricted 24
    - process
      - canceling 73
      - querying 847
    - product version information
      - displaying 6
    - profile 279
    - profile association 273
    - protecting WORM volumes with SnapLock
      - when defining active-data pools 373
      - when defining copy storage pools 364
      - when defining sequential access storage pools 349
    - publications xi

## Q

  - QUERY ACTLOG command 667
  - QUERY ADMIN command 673
  - QUERY ALERTSTATUS command 680
  - QUERY ALERTTRIGGER command 678
  - QUERY ASSOCIATION command 685
  - QUERY AUDITOCCUPANCY command 687
  - QUERY BACKUPSET 690
  - QUERY BACKUPSETCONTENTS 695
  - QUERY CLOPTSET command 698
  - QUERY COLLOCGROUP command 700
  - QUERY CONTENT command 703
  - QUERY COPYGROUP command 710
  - QUERY DATAMOVER command 715
  - QUERY DB command 718
  - QUERY DBSPACE command 721
  - QUERY DEVCLASS command 723
  - QUERY DOMAIN command 728

- QUERY DRIVE command 731
- QUERY DRMEDIA command 735
- QUERY DRMSTATUS command 744
- QUERY ENABLED command 747
- QUERY EVENT command 749
- QUERY EVENTRULES command 761, 764
- QUERY FILESPACE command 772
- QUERY LIBRARY command 780
- QUERY LIBVOLUME command 783
- QUERY LICENSE command 786
- QUERY LOG command 790
- QUERY MACHINE command 792
- QUERY MEDIA command 795
- QUERY MGMTCLASS command 801
- QUERY MONITORSETTINGS command 804
- QUERY MONITORSTATUS command 807
- QUERY MOUNT command 811
- QUERY NASBACKUP command 813
- QUERY NODE command 817
- QUERY NODEDATA command 829
- QUERY OCCUPANCY command 834
- QUERY OPTION command 838
- QUERY PATH command 840
- QUERY POLICYSET command 844
- QUERY PROCESS command 847
- QUERY PROFILE command 851
- QUERY PROXYNODE command 854
- QUERY PVUESTIMATE command 855
- QUERY RECOVERYMEDIA command 859
- query replicate server 881, 1014
- QUERY REPLICATION command 862
- QUERY REPLNODE command 875
- QUERY REPLRULE command 879
- QUERY REPLSERVER command 881
- QUERY REQUEST command 883
- QUERY RESTORE command 884
- QUERY RPFCONTENT command 887
- QUERY RPFIL command 889
- QUERY SCHEDULE command 892
- QUERY SCRATCHPADENTRY 900
- QUERY SCRIPT command 902
- QUERY SERVER command 905
- QUERY SERVERGROUP command 910
- QUERY SESSION command 912
- QUERY SHREDSTATUS command 917
- QUERY SPACETRIGGER command 919
- QUERY SSLKEYRINGPW command 921
- QUERY STATUS command 922
- QUERY STATUSTHRESHOLD command 932
- QUERY STGPOOL command 935
- QUERY SUBSCRIBER command 948
- QUERY SUBSCRIPTION command 950
- QUERY SYSTEM command 952
- QUERY TAPEALERTMSG command 954
- QUERY TOC command 955
- QUERY VIRTUALFSMAPPING command 958
- QUERY VOLHISTORY command 960
- QUERY VOLUME command 968
- QUERYAUTH server option 1539
- querying
  - activity log 667
  - administrator 673
  - audit occupancy 687
  - backup sets 690
  - clients with schedules 685
  - completed events 749
  - contents of a backup set 695

- querying (*continued*)
  - contents of a volume 703
  - copy group 710
  - database 718
  - device class 723
  - domain 728
  - drive 731
  - event server 764
  - file space 772
  - library 780
  - library volume 783
  - license 786
  - management class 801
  - mount requests 883
  - node 817
  - node groups 832
  - occupancy 834
  - option 838
  - policy set 844
  - process 847
  - profile 851
  - recovery log 790
  - recovery plan file 889
  - recovery plan file content 887
  - scheduled events 749
  - schedules 892
  - script 902
  - server group 910
  - session 912
  - space trigger 919
  - SSL key database password 921, 1175
  - status 922
  - storage pool 935
  - storage pool volume 968
  - subscriber 948
  - subscription 950
  - volume history file 960
- QUIT command 975
- quitting
  - batch mode 3
  - interactive mode 2, 975

## R

- random access volumes 330
- raw partitions, performance effect 330
- RECLAIM parameter
  - DEFINE STGPOOL
    - active-data pool 372
    - copy sequential access 363
    - primary sequential access 348
  - UPDATE STGPOOL
    - active-data pool 1429
    - copy sequential access 1422
    - primary sequential access 1410
- RECLAIM STGPOOL command 976
- RECLAIMDELAY option 1540
- RECLAIMPERIOD option 1541
- reclamation of volumes
  - specifying a threshold of reclaimable space
    - using the DEFINE STGPOOL command 348, 363, 372
    - using the UPDATE STGPOOL command 1410, 1422, 1429
  - specifying multiple concurrent processes
    - using the DEFINE STGPOOL command 364, 372
    - using the UPDATE STGPOOL command 1423, 1430
  - starting manually 976

- RECLAMATIONTYPE parameter
  - defining active-data pools 373
  - defining copy storage pools 364
  - defining sequential access storage pools 349
- RECONCILE VOLUMES command 979
- recovery
  - damaged files 989, 1028
- recovery log
  - installing 1594
  - setting options for 1461
- recovery logs 1590
- recovery of damaged files, setting 1158
- recovery plan file
  - display contents 887
  - query information 889
  - set expire days 1130
- recovery plan prefix 1126
- redirecting command output 4
- redirection characters
  - types of 4
  - using 4
- refresh interval
  - setting 1105
- REFRESHSTATE parameter
  - AUDIT LIBRARY 39
- REGISTER ADMIN command 983
- REGISTER LICENSE command 987
- REGISTER NODE command 989
- registering
  - administrator 983
  - license 987
  - node 989
- registration
  - closed 1156
  - description of 1156
  - setting 1156
- REMOVABLEFILE device type 213, 1278
- REMOVABLEFILE, CD support 213
- REMOVE ADMIN command 1008
- REMOVE NODE command 1010
- REMOVE REPLNODE command 1012
- REMOVE REPLSERVER command 1014
- removing
  - administrator 1008
  - client association 390
  - node 1010
- RENAME ADMIN command 1017
- RENAME FILESPACE command 1019
- RENAME NODE command 1023
- RENAME SCRIPT command 1025
- RENAME SERVERGROUP command 1026
- RENAME STGPOOL command 1027
- renaming
  - administrator 1017
  - file space 1019
  - node 1023
  - script 1025
  - server group 1026
  - storage pool 1027
- REPLACEDFS parameter
  - EXPORT ADMIN 488
  - EXPORT NODE 509
  - EXPORT POLICY 517
  - EXPORT SERVER 536
  - IMPORT ADMIN 570
  - IMPORT NODE 576
  - IMPORT POLICY 581
- REPLBATCHSIZE option 1545
- REPLICATE NODE command 1028
- replicating
  - force reconcile 1028
- replication
  - See node replication
- replication server, set failover address 1134
- replication server, setting or removing target 1163
- replication server, verifying policies 1452
- REPLSIZETHRESH option 1546
- REPLY command 1038
- reporting error message for ANR9999D 1104
- REPORTRETRIEVE server option 1544
- REQSYSAUTHOUTFILE option 1547
- request, mount 76, 883
- RESET PASSEXP command 1039
- RESOURCETIMEOUT option 1548
- RESTORE DB, DSMSEV 1604
- RESTORE NODE command 1044
- RESTORE STGPOOL command 1049
- RESTORE VOLUME command 1053
- RESTOREINTERVAL option 1549
- restoring
  - as action on client command 135, 290, 1359
- retention period
  - description of 1133
  - setting 1133
- retention period, event record
  - managing 1133
  - querying 922
- RETENTIONEXTENSION option 1550
- RETONLY parameter
  - DEFINE COPYGROUP, backup 151
  - UPDATE COPYGROUP, backup 1222
- retry period
  - description of 1165
  - setting 1165
- return code checking 1613
- RETVAR parameter
  - DEFINE COPYGROUP, archive 155
  - UPDATE COPYGROUP, archive 1224
- REUSEDELAY parameter
  - DEFINE STGPOOL
    - active-data pool 375
    - copy sequential access 366
    - primary sequential access 352
  - UPDATE STGPOOL
    - active-data pool 1432
    - copy sequential access 1424
    - primary sequential access 1413
- reusing volumes
  - active-data pool 372, 1429
  - copy sequential access 363, 1422
  - primary sequential access 348, 1410
- REVOKE AUTHORITY command 1058
- REVOKE PROXYNODE command 1062
- revoking
  - authority 1058
  - types of privilege classes 1058
- roll-forward recovery log mode 50
- ROLLBACK command 1063
- rolling back commands in a macro 1063
- routing commands 17
- RUN command 1064

## S

- SAN
  - DEFINE LIBRARY command 232
  - UPDATE LIBRARY command 1297
- SAN tape devices 234, 237, 242, 245, 1298, 1300, 1306
- schedule
  - administrative command 300
  - associating with a client node 127
  - client 286, 300
  - copying 108
  - defining 257, 285
  - deleting 434
  - description of 300
  - querying 892
  - querying results of (events) 749
  - restrictions of 286
  - startup window, defining schedule 295, 305
  - startup window, updating schedule 1364, 1373
  - types of 285
  - updating 1354
- schedule event
  - querying 749
  - setting start date for displaying 750
  - setting start time for displaying 751, 752, 757, 759
  - viewing information about 749
- scheduling mode
  - information about 1166
  - setting 1166
- scratch volumes in storage pool
  - defining a storage pool 351, 365, 374
  - updating a storage pool 1413, 1424, 1431
- scripts
  - copying 112
  - defining 312
  - deleting 438
  - querying 902
  - renaming 1025
  - running 1064
  - updating 1379
- SEARCHMPQUEUE option 1551
- Secure Sockets Layer (SSL) 315, 905, 1382
- security options and licensing options 1463
- security, encryption
  - 3592 devices 166, 1234
  - LTO devices 203, 1269
  - StorageTek devices 190, 1257
- SELECT command 1067
- sequential volume history
  - backing up 66
  - deleting 449
  - display sequential volume history 960
  - querying 960
- SERIALIZATION parameter
  - DEFINE COPYGROUP, archive 156
  - DEFINE COPYGROUP, backup 151
  - UPDATE COPYGROUP, archive 1225
  - UPDATE COPYGROUP, backup 1222
- server
  - disabling sessions, inbound and outbound 463
  - enabling sessions, inbound and outbound 473
  - exporting 519
  - importing 582
  - migrating 609
  - setting name for 1171
- server console
  - restrictions 8
  - using 8
- server name settings 1171
- server options
  - 3494SHARED 1465
  - ACSACCESSID 1466
  - ACSLOCKDRIVE 1467
  - ACSQUICKINIT 1468
  - ACSTIMEOUTX 1469
  - ACTIVELOGDIRECTORY 1470
  - ACTIVELOGSIZE 1471
  - ADMINCOMMTIMEOUT 1472
  - ADMINIDLETIMEOUT 1473
  - ADMINONCLIENTPORT 1474
  - ALIASHALT 1475
  - ALLOWREORGINDEX 1476
  - ALLOWREORGTABLE 1477
  - ARCHFAILOVERLOGDIRECTORY 1478
  - ARCHLOGCOMPRESS 1479
  - ARCHLOGDIRECTORY 1480
  - ARCHLOGUSEDTHRESHOLD 1481
  - ASSISTVCRRECOVERY 1482
  - AUDITSTORAGE 1483
  - BEGINREORGTIME 1542, 1543
  - changing with SETOPT command 838
  - CHECKTAPEPOS 1484
  - CLIENTDEDUPTXNLIMIT 1486
  - COMMMETHOD 1488
  - COMMTIMEOUT 1490
  - DBDIAGLOGSIZE 1491
  - DBDIAGPATHFSTHRESHOLD 1493
  - DBMEMPERCENT 1494
  - DBMTCPPORT 1495
  - DEDUPTIER2FILESIZE 1497
  - DEDUPTIER3FILESIZE 1498
  - DEVCONFIG 1499
  - DISABLEREORGTABLE 1500
  - DISABLESCHEDS 1501
  - DISALLOWDES 1502
  - DISPLAYLFINFO 1503
  - DNSLOOKUP 1504
  - DRIVEACQUIRERETRY 1505
  - ENABLENASDEDUP 1506
  - EVENTSERVER 1507
  - EXPINTERVAL 1508
  - EXPQUIET 1509
  - FFDCLOGNAME 1510
  - FFDCMAXLOGSIZE 1511
  - FFDCNUMLOGS 1512
  - FILEEXIT 1513
  - FILETEXTEXIT 1514
  - FSUSEDTHRESHOLD 1515
  - IDENTIFYAUTOSTART 1496
  - IDLETIMEOUT 1516
  - KEEPALIVE 1517
  - KEEPALIVEINTERVAL 1519
  - KEEPALIVETIME 1518
  - LANGUAGE 1520
  - LDAPCACHEDURATION 1522
  - LDAPURL 1523
  - MAXSESSIONS 1524
  - message options 1462
  - MESSAGEFORMAT 1525
  - MIRRORLOGDIRECTORY 1526
  - modifying the file 1457
  - MOVEBATCHSIZE 1527
  - MOVESIZETHRESH 1528
  - MSGINTERVAL 1529
  - NDMPCONNECTIONTIMEOUT 1530



server options (*continued*)

- NDMPCONTROLPORT 1531
- NDMPENABLEKEEPALIVE 1532
- NDMPPORTRANGE 1533
- NDMPREFDATAINTERFACE 1534
- NOPREEMPT 1535
- NORETRIEVEDATE 1536
- NUMOPENVOLSALLOWED 1537
- QUERYAUTH 1539
- querying 1457
- RECLAIMDELAY 1540
- RECLAIMPERIOD 1541
- REPLBATCHSIZE 1545
- REPLSIZETHRESH 1546
- REPORTRETRIEVE 1544
- REQSYSAUTHOUTFILE 1547
- RESOURCETIMEOUT 1548
- RESTOREINTERVAL 1549
- RETENTIONEXTENSION 1550
- SEARCHMPQUEUE 1551
- SERVERDEDUPTXNLIMIT 1552
- SHMPORT 1554
- SHREDDING 1555
- SNMPHEARTBEATINTERVAL 1556
- SNMPMESSAGECATEGORY 1557
- SNMPSUBAGENT 1558
- SNMPSUBAGENTHOST 1559
- SNMPSUBAGENTPORT 1560
- SSLDISABLELEGACYTLS 1561
- SSLFIPSMODE 1562
- SSLHIDELEGACYTLS 1563
- SSLINITTIMEOUT 1564
- SSLTCPADMINPORT 1565
- SSLTCPPOINT 1566
- SSLTLS12 1567
- tailoring 1457
- TCPADMINPORT 1568
- TCPBUFSIZE 1569
- TCPNODELAY 1570
- TCPPOINT 1571
- TCPWINDOWSIZE 1572
- TECBEGINVENTLOGGING 1573
- TECHOST 1574
- TECPOINT 1575
- TECUTF8EVENT 1576
- THROUGHPUTDATATHRESHOLD 1577
- THROUGHPUTTIMETHRESHOLD 1578
- TXNGROUPMAX 1579
- UNIQUEDPTECEVENTS 1580
- UNIQUETECEVENTS 1581
- VOLUMEHISTORY 1584

server scripts

- copying 112
- defining 312
- deleting 438
- querying 902
- renaming 1025
- running 1064
- updating 1379

server storage

- setting options for 1460

server-to-server communications 315

- COMMETHOD option 1488
- shared memory between server and client 1488
- SERVERDEDUPTXNLIMIT option 1552

session

- maximum number scheduled 1142

session (*continued*)

- querying 912
- SET ACCOUNTING command 1080, 1184
- SET ACTLOGRETENTION command 1081
- SET ALERTACTIVEDURATION 1083
- SET ALERTEMAIL 1085
- SET ALERTEMAILFROMADDR 1086
- SET ALERTEMAILSMTPHOST 1087
- SET ALERTEMAILSMTPPORT 1088
- SET ALERTINACTIVEDURATION 1084, 1090
- SET ALERTMONITOR 1091
- SET ALERTSUMMARYTOADMINS 1089
- SET ALERTUPDATEINTERVAL 1092
- SET ARREPLRULEDEFAULT command 1095
- SET AUTHENTICATION command 1097
- SET BKREPLRULEDEFAULT command 1098
- SET CLIENTACTDURATION command 1101
- SET commands 1077
- SET CONFIGMANAGER command 1102
- SET CONFIGREFRESH command 1103
- SET CPUINFOREFRESH command 1105
- SET CROSSDEFINE command 1106
- SET DBRECOVERY command 1107
- SET DEDUPVERIFICATIONLEVEL command 1109
- SET DEFAULTAUTHENTICATION command 1111
- SET DISSIMILARPOLICIES command 1113
- SET DRMACTIVEDATASTGPOOL command 1115
- SET DRMCHECKLABEL command 1116
- SET DRMCMDFILENAME command 1117
- SET DRMCOPYSTGPOOL command 1118
- SET DRMCOURIERNAME command 1119
- SET DRMDBBACKUPEXPIREDAYS command 1120
- SET DRMFILPROCEESS command 1122
- SET DRMINSTRPREFIX command 1123
- SET DRMNOTMOUNTABLENAME command 1125
- SET DRMPLANPREFIX command 1126
- SET DRMPLANVPOSTFIX command 1128
- SET DRMPRIMSTGPOOL command 1129
- SET DRMRPFEXPIREDAYS command 1130
- SET DRMVAULTNAME command 1132
- SET EVENTRETENTION command 1133
- SET FAILOVERHLADDRESS 1134
- SET FAILOVERHLADDRESS command 1134
- SET INVALIDPWLIMIT command 1135
- SET LDAPPASSWORD command 1137
- SET LDAPUSER command 1138
- SET LICENSEAUDITPERIOD command 1140
- SET MAXCMDRETRIES command 1141
- SET MAXSCHEDSESSIONS command 1142
- SET MINPWLENGTH command 1144
- SET MONITOREDSEVERGROUP command 1145
- SET MONITORINGADMIN command 1146
- SET NODEATRISKINTERVAL command 1147
- SET PASSEXP command 1149
- SET QUERYSCHEDPERIOD 1153
- SET RANDOMIZE command 1154
- SET REGISTRATION command 1156
- SET REPLRECOVERDAMAGED command 1158
- SET REPLRETENTION command 1161
- SET REPLSERVER command 1163
- SET RETRYPERIOD command 1165
- SET SCHEDMODES command 1166
- SET SCRATCHPADRETENTION 1168
- SET SCRATCHPADRETENTION command 1168
- SET SERVERHLADDRESS command 1169
- SET SERVERLLADDRESS command 1170
- SET SERVERNAME command 1171
- SET SERVERPASSWORD command 1172

- SET SPREPLRULEDEFAULT command 1173
- SET SSLKEYRINGPW command 1175
- SET STATUSATRISKINTERVAL command 1176
- SET STATUSMONITOR command 1178
- SET STATUSREFRESHINTERVAL command 1180
- SET STATUSSKIPASFAILURE command 1182
- SET SUBFILE command 1184
- SET SUMMARYRETENTION command 1185
- SET TAPEALERTMSG command 1186
- SET TOCLOADRETENTION command 1187
- SET VMATRISKINTERVAL command 1188
- SETARCHIVERETENTIONPROTECTION command 1093
- SETOPT command 1151, 1190
- setting
  - accounting record 1080
  - authentication 1097
  - automatic recovery of damaged files 1158
  - configuration manager 1102
  - configuration refresh 1103
  - cross define of a server 1106
  - data deduplication verification level 1109
  - frequency for client-polling 1153
  - frequency for one-time client action 1101
  - high level address of a server 1169
  - license audit period 1140
  - low level address of a server 1170
  - maximum retries of a command 1141
  - maximum scheduled sessions 1142
  - password for a server 1172
  - password for expiration date 1149
  - randomization of start times 1154
  - registration 1156
  - retention period for activity log 1081
  - retention period for event records 1133
  - retry period 1165
  - scheduling mode 1166
  - server name 1171
  - summary retention days 1185
- setting communication options 1458
- shared library type 249, 1309
- SHMPORT option
  - specifying a TCP/IP port address 1554
  - using shared memory 1554
- SHRED DATA command 1192
- shredding data, storage pools
  - backing up 64
  - defining 340
  - moving data 615
  - updating 1403
- SHREDDING option 1555
- simultaneous write to copy storage pools
  - defining random access storage pools 338
  - defining sequential access storage pools 356
  - updating random access storage pools 1401
  - updating sequential access storage pools 1417
- single-instance store (data deduplication) 358, 367, 368, 376, 1419, 1425, 1432, 1433
- SKIPPARTIAL parameter, AUDIT VOLUME 46
- SnapLock for managing WORM FILE volumes
  - when defining active-data pools 373
  - when defining copy storage pools 364
  - when defining sequential access storage pools 349
- SNMPHEARTBEATINTERVAL option 1556
- SNMPMESSAGECATEGORY option 1557
- SNMPSUBAGENT option 1558
- SNMPSUBAGENTHOST option 1559
- SNMPSUBAGENTPORT option 1560
- space management
  - DEFINE MGMTCLASS 257
  - description of 257
  - UPDATE MGMTCLASS 1317
- space trigger commands
  - DEFINE SPACETRIGGER 323
  - DELETE SPACETRIGGER 441
  - QUERY SPACETRIGGER 919
  - UPDATE SPACETRIGGER 1387
- SPACEMGTECHNIQUE parameter
  - DEFINE MGMTCLASS 258
  - UPDATE MGMTCLASS 1317
- SSL (Secure Sockets Layer) 315, 905, 1382
- SSL key database password
  - querying 921
  - setting 1175
- SSLDISABLELEGACYTLS option 1561
- SSLFIPSMODE option 1562
- SSLHIDELEGACYTLS option 1563
- SSLINITTIMEOUT option 1564
- SSLTCPADMINPORT server option 1565
- SSLTCPSPORT server option 1566
- SSLTLS12 option 1567
- stand-alone mode 1586
- start time
  - information about 285
  - setting 1154
- STARTDATE parameter
  - DEFINE SCHEDULE, administrative command 305
  - DEFINE SCHEDULE, client 295
  - UPDATE SCHEDULE, administrative command 1373
  - UPDATE SCHEDULE, client 1364
- STARTTIME parameter
  - DEFINE SCHEDULE, administrative command 305
  - DEFINE SCHEDULE, client 295
  - UPDATE SCHEDULE, administrative command 1373
  - UPDATE SCHEDULE, client 1364
- status
  - event completion 755
  - information about 922
  - querying 922
- STGPOOL parameter, MOVE DATA 615
- STGPOOLS parameter
  - GRANT AUTHORITY 556
  - REVOKE AUTHORITY 1060
- storage 1460
- storage area network (SAN)
  - DEFINE LIBRARY command 232
  - UPDATE LIBRARY command 1297
- storage pool
  - collocating
    - active-data pool 371, 1428
    - copy sequential access 362, 1421
    - primary sequential access 350, 1411
  - defining 330
  - deleting 444
  - migrating 609
  - querying 935
  - reclaiming 976
  - restoring 1049
  - updating 1393
- storage pool volume
  - auditing 43
  - defining 381
  - deleting 454
  - querying 703, 968
  - scratch, active-data pool 374, 1431

- storage pool volume (*continued*)
  - scratch, copy sequential access 365, 1424
  - scratch, primary sequential access 351, 1413
  - updating 1438
  - varying 1455
- storage pools 323
- storage volumes
  - naming 14
  - performance increase with raw partitions 330
- StorageTek drives
  - logical block protection 190
- SWAP parameter, CHECKIN LIBVOLUME 84
- syntax diagram
  - abbreviations 9
  - default value 10
  - fragments 12
  - optional choice 10
  - repeatable choice 11
  - repeating values 11
  - required parameters 9
  - symbols 11
  - using 9
  - variables 10
- SYSCONFIG command (on NAS file server) 262
- system privilege class
  - administrative commands 20

## T

- tape
  - AUDIT LIBRARY command 38
  - AUDIT LIBVOLUME command 40
  - AUDIT VOLUME command 43
  - CHECKIN LIBVOLUME command 79
  - CHECKOUT LIBVOLUME command 86
  - DEFINE DEVCLASS command 161
  - DEFINE LIBRARY command 232, 234, 237, 240, 243, 245, 249
  - DEFINE STGPOOL command 330, 342, 360
  - DEFINE VOLUME command 381
  - DELETE DEVCLASS command 406
  - DELETE DRIVE command 408
  - DELETE LIBRARY command 418
  - DELETE VOLUME command 454
  - LABEL LIBVOLUME command 591
  - QUERY DEVCLASS command 723
  - QUERY DRIVE command 731
  - QUERY LIBRARY command 780
  - QUERY LIBVOLUME command 783
  - QUERY VOLUME command 968
  - storage area network (SAN) 234, 237, 242, 245, 1298, 1300, 1306
  - UPDATE DEVCLASS command 1229
  - UPDATE LIBRARY command 1297, 1298, 1300, 1302, 1304, 1306, 1309
  - UPDATE LIBVOLUME command 1313
  - UPDATE VOLUME command 1438
- tape volume
  - auditing 40
- tape-device encryption
  - 3592 166, 1234
  - ECARTRIDGE 190, 1257
  - LTO 203, 1269
- target replication server policies, enable 1113
- target replication server, enable policies 1113
- target replication server, setting or removing 1163

- TCP keepalive
  - enabling 1532
- TCPADMINPORT server option 1568
- TCPBUFSIZE option 1569
- TCPNODELAY option 1570
- TCPPORT option 1571
- TCPWINDOWSIZE option 1572
- TECBEGINEVENTLOGGING option 1573
- TECHOST option 1574
- TECPORT option 1575
- TECUTF8EVENT option 1576
- threshold, migration
  - random-access storage pools
    - high threshold 335, 1398
    - low threshold 335, 1398
  - sequential-access storage pools
    - high threshold 347, 1409
    - low threshold 348, 1410
- THROUGHPUTDATATHRESHOLD option 1577
- THROUGHPUTTIMETHRESHOLD option 1578
- time interval, setting for checking in volumes 1252, 1269
- timeout, TCP
  - specifying connection timeout 1530
- TOCDestination parameter 152, 1223
- TOSERVER parameter
  - EXPORT ADMIN 488
  - EXPORT NODE 509
  - EXPORT POLICY 517
  - EXPORT SERVER 536
- transfer data by data mover 158
- troubleshooting LAN-free data movement 1443
- TXNGROUPMAX option 1579
- type, device
  - 3590 162, 1230
  - 3592 166, 1234
  - 4MM 172, 1240
  - 8MM 176, 1244
  - CENTERA 182, 1250
  - DLT 184, 1252
  - ECARTRIDGE 190, 1257
  - FILE 197, 216, 1263, 1280
  - GENERICTAPE 200, 1267
  - LTO 203, 204, 1269
  - REMOVABLEFILE 213, 1278
  - SERVER 216, 1280
  - VOLSAFE 218, 1282
- typographic conventions xi

## U

- Ultrium, LTO device type
  - logical block protection 203, 1269
- UNIQUETDPTCEVENTS option 1580
- UNIQUETECEVENTS option 1581
- UNLOCK ADMIN command 1196
- UNLOCK NODE command 1198
- UNLOCK PROFILE command 1200
- unlocking
  - administrator 1196
  - node 1198
  - profile 1200
- Update a 349X library 1298
- Update a file library 1303
- Update a manual library 1304
- Update a SCSI library 1306
- Update a shared library 1309
- Update a VTL library 1310

- UPDATE ADMIN command 1207
- UPDATE ALERTSTATUS 1205
- UPDATE ALERTTRIGGER command 1202
- Update an ACSLS library 1300
- UPDATE BACKUPSET 1211
- UPDATE CLIENTOPT command 1216
- UPDATE CLOPTSET command 1217
- UPDATE COLLOGGROUP command 1218
- UPDATE COPYGROUP command 1219
- UPDATE DATAMOVER command 1227
- UPDATE DEVCLASS command 1229
- UPDATE DOMAIN command 1286
- UPDATE DRIVE command 1288
- UPDATE FILESPACE command 1292
- UPDATE LIBRARY command 1297, 1298, 1300, 1302, 1303, 1304, 1306, 1309, 1310
- UPDATE LIBVOLUME command 1313
- UPDATE MACHINE command 1315
- UPDATE MGMTCLASS command 1317
- UPDATE NODE command 1320
- UPDATE PATH command 1339, 1340, 1344
- UPDATE POLICYSET command 1347
- UPDATE PROFILE command 1349
- UPDATE RECOVERYMEDIA command 1350
- UPDATE REPLRULE command 1352
- UPDATE SCHEDULE command 1354
- UPDATE SCRATCHPADENTRY 1378
- UPDATE SCRIPT command 1379
- UPDATE SERVER command 1382
- UPDATE SERVERGROUP command 1386
- UPDATE SPACETRIGGER command 1387
- UPDATE STATUSTHRESHOLD command 1389
- UPDATE STGPOOL command 1393
- UPDATE VIRTUALFSMAPPING command 1434
- UPDATE VOLHISTORY command 1436
- UPDATE VOLUME command 1438
- updating
  - administrator 1207
  - archive copy group 1224
  - backup copy group 1220
  - backup set 1211
  - copy group 1219
  - device class 1229
  - domain 1286
  - drive 1288
  - file library 1303
  - library 1297, 1298, 1300, 1302, 1304, 1306, 1309
  - library volume 1313
  - management class 1317
  - node 1320
  - node group 1338
  - policy set 1347
  - profile 1349
  - retention period for a backup set 1211
  - schedule 1354
  - script 1379
  - server group 1386
  - storage pool 1393
  - storage pool space triggers 1387
  - storage pool volume 1438
  - TOCDestination copy group parameter 1223
  - volume history 1436
  - VTL library 1310
- USEREXIT option 1582
- using
  - administrative client options 5
  - command-line interface 1

- using (*continued*)
  - continuation characters 13
  - macros 607
  - redirection 4
  - redirection characters 4
  - syntax diagram 9
  - system date on the server 30

## V

- VALIDATE LANFREE command 1443
- VALIDATE POLICYSET command 1445
- VALIDATE REPLICATION command 1447
- VALIDATE REPLPOLICY command 1452
- validating
  - LAN-free 1443
  - policies 1452
  - policy set 1445
- VARY command 1455
- varying volumes 1455
- VERDELETED parameter
  - DEFINE COPYGROUP, backup 150
  - UPDATE COPYGROUP, backup 1221
- VEREXISTS parameter
  - DEFINE COPYGROUP, backup 150
  - UPDATE COPYGROUP, backup 1221
- verify automated library inventory 38
- virtual tape library 250
- VOLSAFE
  - device type 218, 1282
  - labeling volumes 596
- volume
  - labeling library 591
- volume history
  - deleting 449
  - querying 960
- VOLUMEHISTORY option 1584
- volumes
  - reclaiming 976
- volumes, storage
  - defining 381
  - limiting offsite volumes to be reclaimed
    - using the DEFINE STGPOOL command 365, 373
    - using the UPDATE STGPOOL command 1423, 1430
  - naming 14
  - reclaiming
    - from active-data pool 372, 1429
    - from copy sequential access 363, 1422
    - from primary sequential access 348, 1410
  - restoring 1053
  - scratch, maximum in active-data pool 374, 1431
  - scratch, maximum in copy sequential access 365, 1424
  - scratch, maximum in primary sequential access 351, 1413
- VTL
  - DEFINE LIBRARY command 250
  - UPDATE LIBRARY command 1310
- VTL library
  - defining 250
  - updating 1310

## W

- WAITTIME parameter
  - CHECKIN LIBVOLUME command 84
- WHEREACCESS parameter, UPDATE VOLUME 1440
- WHERESTATUS parameter, UPDATE VOLUME 1440

WHERESTGPOOL parameter, UPDATE VOLUME 1440  
wildcard  
    match-any examples 15  
    match-exactly-one examples 15  
    using 14  
WORM devices and media  
    DLT 184  
    IBM 3592 166  
    LTO 203  
    SnapLock for managing WORM FILE volumes 349, 364,  
        373  
    StorageTek ECARTRIDGE 190

## **Z**

ZOSMEDIA 234, 237, 240, 249, 1298, 1300, 1302, 1309







Product Number: 5608-E01  
5608-E02  
5608-E03

Printed in USA